



**José Maria Eduardo
Samuco**

Algoritmos de Otimização Contínua Univariada



**José Maria
Eduardo Samuco**

Algoritmos de Otimização Contínua Univariada

Dissertação apresentada na Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática e Aplicações, especialização em Estatística e Investigação Operacional, realizada sob a orientação científica do Doutor Jorge Manuel Sá Esteves, Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro.

o júri / the jury

presidente / president

Professora Doutora Isabel Maria Simões Pereira

Professora Auxiliar do Departamento de Matemática da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Maria Teresa Torres Monteiro

Professora Auxiliar do Departamento de Produção e Sistemas da Universidade do Minho

Professor Doutor Jorge Manuel Sá Esteves

Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

Agradeço
ao Professor Doutor Jorge Sá Esteves, pela disponibilidade manifestada
e pela forma cuidada, sábia e rigorosa como me orientou na elaboração
desta dissertação.

Palavras-chave

otimização, algoritmos, função unimodal, função convexa, eliminação, interpolação, intervalo de incerteza.

Resumo

Nesta dissertação são estudados alguns métodos numéricos de otimização de funções reais contínuas de uma variável real. Nesse sentido, e antes desta abordagem, são analisadas as técnicas clássicas de otimização, sendo feito um estudo de condições de otimalidade de funções convexas e de funções contínuas. O estudo dos métodos numéricos é dividido em três categorias: métodos intervalares de eliminação (método de busca dicotômica, método de busca por bissecção, método de Fibonacci e método da secção áurea), métodos de aproximação polinomial (método de interpolação quadrática, método de interpolação cúbica e algoritmo de Davies, Swann e Campey) e busca linear inexata. Os métodos aplicam-se a funções unimodais, razão pela qual este conceito é introduzido e é discutida a sua utilização na redução de intervalos de incerteza. No final do estudo de cada método, são apresentados problemas resolvidos, com indicação de todos os passos de cada iteração ou aplicando rotinas em MATLAB, cujos códigos são explicitados ao longo do texto. Estudamos também propriedades dos números de Fibonacci para a verificação de que, em certo sentido, o método de Fibonacci é o método ideal para a contração do intervalo. Essas propriedades permitem também verificar a estreita inter-relação entre este método e o método da secção áurea.

Keywords

optimization, algorithms, unimodal function, convex function, elimination, interpolation, interval of uncertainty.

Abstract

In this dissertation some numerical methods for optimization of continuous real functions of a single real variable are studied. In this sense, and before this approach, the classical optimization techniques are analyzed and a study of optimality conditions for convex functions and continuous ones is made. The study of numerical methods is divided into three categories: interval methods of elimination (dichotomous search method, interval halving method, Fibonacci's method and golden section method), polynomial approximation methods (quadratic interpolation method, cubic interpolation method and Davies, Swann and Campey's algorithm) and inexact line search. The methods are applied to unimodal functions, that is why this concept is introduced and its use in reducing uncertainty intervals is discussed. For each method, solved problems are presented, showing all steps of each iteration or applying routines in MATLAB, whose codes are specified throughout the text. Properties of Fibonacci numbers are also studied showing that, in a sense, the Fibonacci method is the optimal method for contraction of the interval. These properties allow us also check the close interrelationship between this method and the method of golden section.

“Optimization is the art, science, and mathematics of finding the “best” member of a finite or infinite set of possible choices, based on some objective measure of the merit of each choice in the set.”

John Lawrence Nazareth
(*Professor Emeritus, Washington*
State University)

Conteúdo

| | |
|---|-----------|
| Conteúdo | i |
| Introdução | 1 |
| 1 Fundamentos da Otimização Unidimensional | 6 |
| 1.1 Noções Elementares | 6 |
| 1.2 Otimização de Funções Contínuas | 8 |
| 1.3 Otimização de Funções Convexas | 12 |
| 1.4 Otimização Clássica | 20 |
| 1.5 Aplicações | 23 |
| 2 Métodos Intervalares de Eliminação | 31 |
| 2.1 Funções Unimodais | 31 |
| 2.2 Método de Busca Dicotómica | 34 |
| 2.3 Método de Busca por Bissecção | 46 |
| 2.4 Método de Fibonacci | 55 |
| 2.5 Método da Secção Áurea | 66 |
| 2.6 Comparação de Métodos de Eliminação | 73 |
| 3 Métodos de Aproximação Polinomial | 80 |
| 3.1 Método de Interpolação Quadrática | 81 |
| 3.1.1 Interpolação em Três Pontos | 81 |
| 3.1.2 Interpolação em Dois Pontos | 92 |
| 3.2 Método de Interpolação Cúbica | 96 |
| 3.3 Método de Davies, Swann e Campey | 109 |

| | |
|---|------------|
| 4 Busca Linear Inexata | 123 |
| 4.1 Condições de Goldstein | 123 |
| 4.2 Algoritmo de Busca Linear Inexata de Fletcher | 128 |
| Conclusões | 138 |
| Apêndices | 143 |
| A Propriedades dos Números de Fibonacci | 144 |
| B Determinante da Matriz de Vandermonde | 155 |
| Bibliografia | 159 |

Introdução

Enquadramento

A otimização é um processo intrínseco à natureza. Por exemplo, os sistemas físicos tendem ao estado de energia mínima e os raios de luz seguem um caminho que minimiza o tempo de percurso. As moléculas num sistema químico isolado reagem umas com as outras até que a energia potencial total dos seus eletrões seja mínima. Também o Homem procura permanentemente otimizar as suas condições de vida. As companhias aéreas selecionam tripulações e aeronaves para minimizar o custo das suas operações. As empresas procuram o lucro máximo e o prejuízo mínimo.

A otimização é uma ferramenta importante para a tomada de decisão durante a análise e o projeto de sistemas físicos. Para utilizar esta ferramenta torna-se necessário identificar o objetivo, ou seja, a grandeza que irá medir a *performance* do sistema, dita função objetivo. Por exemplo, a energia, o custo, a potência, o tempo, a temperatura, etc. O objetivo depende das características do sistema, expressas nas variáveis de decisão, as quais podem estar sujeitas a restrições, limites impostos pelo sistema em estudo. Portanto, a otimização visa encontrar o valor das variáveis de decisão que otimizam o objetivo.

A identificação dos objetivos, das variáveis de decisão e das restrições constituem a modelização do problema. Trata-se de traduzir matematicamente uma situação real. Deste modo, o processo de modelização é determinante na resolução de um problema de otimização. Se o modelo é simplista demais, poderá não dar informações úteis sobre o problema prático, mas se for muito complexo, pode tornar-se demasiado difícil de resolver.

Uma vez que não há um algoritmo universal para resolver todos os problemas, depois da modelização, segue-se a escolha de um método para resolver o problema, o qual poderá requerer a utilização de ferramentas computacionais. Existem numerosos algoritmos, cada um dos quais está adaptado para um tipo particular de problemas de otimização.

A escolha do algoritmo é muito importante, pois pode determinar se o problema é resolvido rápida ou lentamente, e se, de facto, a solução é encontrada.

Matematicamente falando, a otimização é a minimização ou maximização de uma função sujeita a restrições nas suas variáveis.

Os problemas de otimização de que trataremos podem então ser escritos da seguinte forma:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in [a, b] \subset \mathbb{R}, \end{array}$$

sendo f uma função real definida num intervalo fechado. Caso pretendamos maximizar f , podemos resolver um problema equivalente que corresponde a minimizar $-f$.

Otimização Discreta e Otimização Contínua

Em alguns problemas de otimização as variáveis só fazem sentido se elas assumirem valores inteiros. Estes problemas são conhecidos como problemas de programação inteira.

O termo genérico *otimização inteira* geralmente refere-se a problemas em que a solução que procuramos é uma de uma série de objetos pertencente a um conjunto finito. Por oposição, nos problemas de otimização contínua, pretende-se encontrar uma solução a partir de um conjunto infinito não numerável. Os problemas que aqui trataremos são desta classe. Os problemas de otimização contínua são normalmente mais fáceis de resolver, porque o comportamento das funções torna possível usar informações da função objetivo e das restrições no ponto x e depois deduzir informações sobre o comportamento da função numa vizinhança de x . A mesma afirmação não pode ser feita sobre os problemas discretos, onde os pontos que estão “perto” em certo sentido, podem ter valores da função marcadamente diferentes. Além disso, o conjunto de soluções possíveis é demasiado grande para fazer uma busca exaustiva procurando o melhor valor neste conjunto finito.

Otimização Restrita e não Restrita

Os problemas de otimização podem ser classificados de acordo com a natureza da função objetivo e das restrições (lineares, não lineares, convexas), o número de variáveis (pequeno ou grande), a *suavidade* das funções (diferenciáveis ou não diferenciáveis), e assim por diante. Possivelmente, a distinção mais importante é entre os problemas que têm restrições sobre as variáveis e os que não o têm.

Problemas de otimização sem restrições surgem diretamente em muitas aplicações práticas. Se existem limitações naturais sobre as variáveis, às vezes é seguro ignorá-las e supor que elas não têm efeito sobre a solução ótima. Surgem problemas sem restrições também como reformulações de problemas de otimização restrita, na qual as restrições são substituídas por termos de penalização na função objetivo que têm o efeito impedir violações das restrições.

Problemas de otimização restrita surgem a partir de modelos que incluem restrições explícitas sobre as variáveis. Estas restrições podem ser limites simples (“*box constraints*”),

como $0 \leq x \leq 100$, ou desigualdades não lineares que representam complexas relações entre as variáveis.

Os problemas de programação não linear, em que pelo menos algumas das restrições ou a função objetivo são funções não lineares, tendem a surgir naturalmente em ciências (Física, Química, Biologia) e em engenharia, e são extensivamente utilizados na Gestão e nas ciências económicas.

Otimização Local e Global

Os algoritmos de otimização mais rápidos destinam-se apenas a uma solução local, um ponto em que a função objetivo é menor do que em todos os outros pontos possíveis da sua vizinhança. Esses algoritmos nem sempre procuram o melhor de todos esses mínimos, ou seja, a solução global. As soluções globais são necessárias (ou, pelo menos, altamente desejáveis) em algumas aplicações, mas são normalmente difíceis de identificar e ainda mais difíceis de localizar. Um caso especialmente importante é a programação convexa, em que todas as soluções locais são também soluções globais. No entanto, os problemas gerais não lineares, tanto com ou sem restrições, podem possuir soluções locais que não são soluções globais.

Algoritmos de Otimização

Os algoritmos de otimização são normalmente iterativos. Estes processos partem de uma estimativa inicial do valor ótimo da variável e geram uma sequência de estimativas melhoradas até se obter uma solução que satisfaça um critério de paragem.

A estratégia utilizada para ir de uma iteração para a próxima distingue um algoritmo de outro. A maior parte das estratégias faz uso de valores da função objetivo f , das restrições e, eventualmente, das primeira e segunda derivadas de f . Alguns algoritmos acumulam informações obtidas em iterações anteriores, enquanto outros usam apenas informações locais do ponto atual. Independentemente dessas especificidades, todos os bons algoritmos devem possuir as seguintes propriedades:

- **Consistência Analítica:** Devem estar sustentados por provas teóricas de convergência suportadas por algumas hipóteses o mais gerais possível. Uma prova forte é, por exemplo, a da precisão arbitrária ser atingível se não houver erros de arredondamento. Noutros casos, existe apenas uma prova que garante uma dada precisão máxima sob certas condições. Estas provas de consistência analítica baseiam-se em supor que lidamos com variáveis reais;
- **Estabilidade Numérica:** Os cálculos não devem acumular os erros de arredondamento de forma a que a precisão do resultado seja destruída. Deve supor-se que o

problema a tratar é bem condicionado porque, noutros casos, é difícil calcular uma solução viável pois esta é muito sensível aos dados iniciais do problema;

- **Convergência da Implementação:** Se um método consistente for implementado (usando um formato de vírgula flutuante) e se revelar estável então diz-se convergente;
- **Eficiência Computacional:** Não devem exigir muito tempo computacional ou muito espaço de armazenamento na memória;
- **Robustez e Flexibilidade:** Devem ter uma boa *performance* numa grande variedade de problemas da sua classe, para todas as opções razoáveis dos valores iniciais;
- **Simplicidade:** Preferem-se sempre os métodos fáceis de perceber, programar e calibrar.

Estes objetivos podem entrar em conflito. Por exemplo, um método rapidamente convergente para programação não linear pode exigir muito espaço de memória em grandes problemas. Por outro lado, um método robusto pode ser lento.

Objetivos Gerais da Dissertação

O grande propósito desta dissertação é apresentar uma abordagem integrada de modelos de otimização local contínua univariada. Não exigiremos que a função seja convexa bastando que seja unimodal. Em quase todos os casos trataremos de otimização restrita a um intervalo fechado (compacto de \mathbb{R}). Por vezes exigiremos que a função seja diferenciável mas isso não acontece nos métodos intervalares. Nesse contexto, podemos então elencar as seguintes metas a atingir:

- Estudar métodos numéricos de otimização de funções unimodais num intervalo fechado;
- Comparar métodos intervalares de eliminação e, em especial, as relações entre os métodos de Fibonacci e da secção áurea;
- Apresentar os métodos de interpolação quadrática e cúbica como algoritmos alternativos possíveis;
- Mostrar o algoritmo de Davies, Swann e Campey como exemplo de método híbrido (intervalar/interpolador);
- Aflorar as possibilidades de definir métodos de busca inexata muito usados em *line search* na otimização multivariada;
- Modelar situações práticas formulando problemas de otimização contínua univariada tendo algum cuidado no interesse didático deste assunto;

- Estudar problemas de otimização de funções contínuas unimodais, com uma variável, usando uma abordagem algorítmica e computacional, testada para vários problemas resolvidos;
- Usar o MATLAB para implementação computacional de algoritmos e como ferramenta auxiliar na tomada de decisões.

Organização da Dissertação

No Capítulo 1, para além de noções elementares e das técnicas clássicas de otimização, é feito um estudo de condições de otimalidade de funções convexas univariadas. No final do capítulo, são apresentados vários problemas resolvidos, como aplicação das técnicas clássicas de otimização.

O capítulo seguinte é dedicado aos métodos intervalares de eliminação. Depois de introduzida a noção de função unimodal, são estudados os métodos de busca dicotômica, de busca por bissecção, de Fibonacci e da secção áurea. No final do estudo de cada método são apresentados problemas resolvidos, com indicação de todos os passos de cada iteração ou são aplicadas rotinas em MATLAB. O capítulo termina com uma comparação da eficiência dos métodos analisados.

O Capítulo 3 é dedicado aos métodos de aproximação polinomial quadrática e cúbica. Adicionalmente é feita uma abordagem do algoritmo de busca de Davies, Swann e Campey, o qual combina a interpolação polinomial com a eliminação intervalar.

O capítulo final relativo às conclusões é precedido do Capítulo 4, onde é analisada uma classe de métodos com muita importância para vários métodos de otimização multi-dimensional. Trata-se da busca linear inexata, sendo discutido o algoritmo devido a Roger Fletcher [14].

A dissertação termina com dois apêndices. No Apêndice A são estudadas propriedades dos números de Fibonacci, relevantes para a compreensão do método de Fibonacci e para o relacionar com o método da secção áurea. O estudo destas propriedades conduz à prova de que o método de Fibonacci é um método de otimização ótimo. No Apêndice B, depois da definição de matriz de Vandermonde são provadas condições que garantem quando esta não é singular. Esta propriedade é importante para o estudo dos métodos de interpolação polinomial.

Capítulo 1

Fundamentos da Otimização Unidimensional

Um problema de otimização pode não ter solução. Há vários resultados da análise matemática que garantem a não impossibilidade de um problema de otimização — *Teorema de Weierstrass* — ou a forma de obter os candidatos a solução do problema — *Teorema de Fermat* — e condições suficientes para que os candidatos obtidos por aplicação do *Teorema de Fermat* sejam maximizantes ou minimizantes. Os métodos clássicos da análise matemática usam derivadas e o valor da função objetivo, dependendo de condições de diferenciabilidade desta e da resolução algébrica da equação $f'(x) = 0$, onde f é a função objetivo.

Vamos de seguida fazer uma abordagem dos métodos clássicos de otimização em \mathbb{R} e deixaremos para mais tarde a abordagem dos métodos numéricos de otimização, baseados em algoritmos iterativos. Neste capítulo, teremos como referência [1, Antoniou e Lou], [3, Cardoso], [9, Carvalho], [5, Chong e Zak], [10, Sousa Pinto], [11, Sá Esteves], [13, Figueira], [31, Paressini, Sullivan e Uhl Jr], [26, Rao], [27, Santos], [30, Stewart] e [32, Swokowski].

1.1 Noções Elementares

Dada uma função real de variável real f e $\mathcal{A} \subseteq \mathbb{R}$, estudaremos os problemas de otimização unidimensional que se podem formular genericamente do modo indicado a seguir em (1.1).

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in \mathcal{A}. \end{array} \tag{1.1}$$

A função f , que pretendemos minimizar, denomina-se por *função objetivo*. À variável independente da função objetivo dá-se o nome de *variável de decisão*. As condições a que deve obedecer a variável de decisão designam-se por *restrições* e \mathcal{A} é a *região admissível*.

O Problema (1.1) pode ser visto como um problema de decisão que envolve determinar o “melhor” valor $x \in \mathcal{A}$ da variável de decisão. Por “melhor” valor de x deve entender-se o que produz o menor valor da função objetivo. Diz-se que este valor é um minimizante de f em \mathcal{A} . O mínimo se existir é único, mas é possível que haja vários minimizantes. Neste caso, determinar um minimizante é considerado suficiente para resolver o problema.

Há problemas de otimização que requerem a maximização da função objetivo. No entanto, esses problemas podem ser representados na forma do Problema (1.1), uma vez que maximizar f é equivalente a minimizar $-f$. O valor x^* que maximiza $f(x)$ é o mesmo que minimiza $-f(x)$ e verifica-se que $\max f(x) = -\min (-f(x))$, como se mostra na Figura 1.1. Assim, a formulação apresentada para o Problema (1.1) é válida também para problemas de maximização. Sendo assim, pode-se, sem perda de generalidade, considerar apenas problemas de minimização. É o que faremos ao longo de todo o trabalho. O Pro-

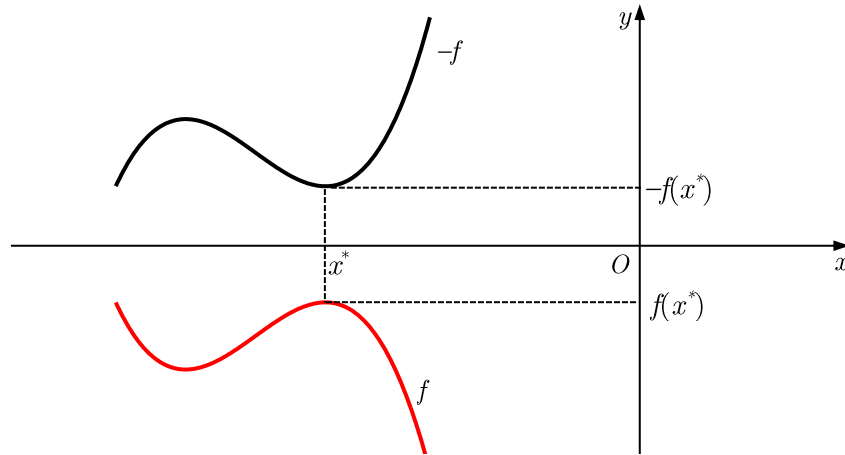


Figura 1.1: O maximizante de f é igual ao minimizante de $-f$.

blema (1.1) é a forma geral de um *problema de otimização unidimensional com restrições*, dado que a variável de decisão está restringida ao conjunto \mathcal{A} . No caso de $\mathcal{A} = \mathbb{R}$, diz-se que o Problema (1.1) é um *problema de otimização unidimensional sem restrições*. O conjunto de relações matemáticas (equações, inequações, etc.) que definem \mathcal{A} denomina-se por *conjunto de restrições*.

Atendendo ao Problema (1.1), distinguiremos dois tipos de minimizantes, como se especifica nas duas definições seguintes.

Definição 1.1 (minimizante local) *Seja $f : \mathcal{A} \subset \mathbb{R} \rightarrow \mathbb{R}$. Um ponto $x^* \in \mathcal{A}$ diz-se um minimizante local de f se existir um intervalo aberto I centrado em x^* tal $f(x^*) \leq f(x), \forall x \in I \cap \mathcal{A}$. Diz-se, então, que $f(x^*)$ é um mínimo local de f .*

Definição 1.2 (minimizante absoluto) *Seja $f : \mathcal{A} \subset \mathbb{R} \rightarrow \mathbb{R}$. Um ponto $x^* \in \mathcal{A}$ diz-se um minimizante absoluto (ou global) de f em \mathcal{A} se $f(x^*) \leq f(x), \forall x \in \mathcal{A}$. Diz-se, então, que $f(x^*)$ é o mínimo absoluto (ou global) de f em \mathcal{A} .*

Se nas definições 1.1 e 1.2, considerarmos desigualdades em sentido estrito, os minimizantes designam-se por minimizantes estritos. Teríamos, então, um *minimizante local estrito*, na Definição 1.1, e um *minimizante global estrito*, na Definição 1.2, sendo necessário acrescentar, nesta definição, $x \neq x^*$.

Na Figura 1.2 exemplificam-se graficamente as definições anteriores. Na função g , todos os pontos do intervalo $[2,3]$ são minimizantes não estritos e 4 é um minimizante estrito da função h .

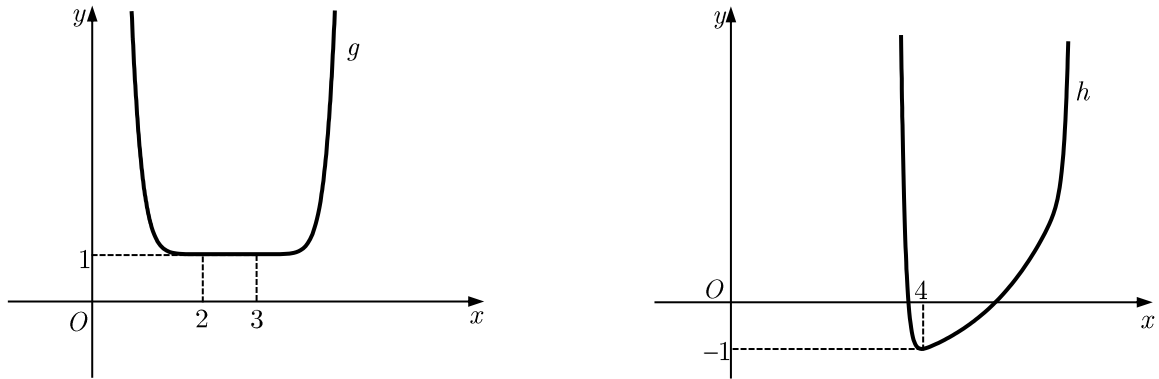


Figura 1.2: 2 e 3 são minimizantes não estritos da função g e 4 é um minimizante estrito da função h .

Dada uma função real de variável real f , de domínio \mathcal{A} , com um único minimizante, a expressão $\arg \min f(x)$ representa um valor $x^* \in \mathcal{A}$ que minimiza $f(x)$. Tomando como exemplo a função f representada graficamente na Figura 1.3, $\arg \min f(x) = 3$. No caso de se escrever $\arg \min_{x \in \mathcal{A}} f(x)$, deve-se assumir que \mathcal{A} é um subdomínio de f . Para a função que estamos a analisar, $\arg \min_{x \leq 1} f(x) = 1$. Em geral, caso exista e seja único, consideramos $\arg \min_{x \in \mathcal{A}} f(x)$ como sendo o minimizante global de f em \mathcal{A} .

1.2 Otimização de Funções Contínuas

As funções contínuas são uma classe de funções alvo neste trabalho, razão pela qual vamos tratar de algumas propriedades fundamentais dessas funções. A propriedade seguinte garante que uma função definida num intervalo fechado e limitado, contínua nesse intervalo, não pode passar de um valor para outro sem passar pelos valores intermédios.

Teorema 1.1 (Teorema de Bolzano) *Sejam $f : [a, b] \rightarrow \mathbb{R}$ uma função ($a < b$) e $f(a) \neq f(b)$. Se f é contínua em $[a, b]$ e ou $f(a) < y_0 < f(b)$ ou $f(b) < y_0 < f(a)$, então existe pelo menos um $c \in]a, b[$ tal que $f(c) = y_0$.*

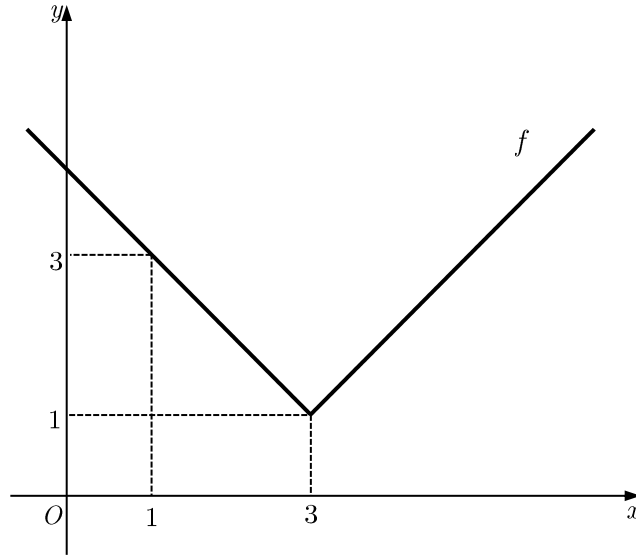


Figura 1.3: $\arg \min f(x) = 3$ e $\arg \min_{x \leq 1} f(x) = 1$.

Prova: Admitamos, sem perda de generalidade, que $f(a) < f(b)$. Consideremos o conjunto $X = \{x \in [a, b] : f(x) < y_0\}$.

Uma vez que, por hipótese, $f(a) < y_0$, temos que $a \in X$ e, portanto, X é um conjunto não vazio. Atendendo a que b é um majorante de X , conclui-se que X é um conjunto majorado. Logo, pelo Axioma de Supremo¹, X admite supremo.

Designemos o supremo de X por c . Então, $c \in [a, b]$. Vamos provar que $f(c) = y_0$.

Com efeito, se fosse $f(c) < y_0$, seria decerto $c < b$, visto que $f(b) > y_0$, por hipótese. Uma vez que f é contínua em c e $f(c) < y_0$, existe $\delta > 0$ tal que, para todo o $x \in [a, b] \cap V_\delta(c)$, $f(x) < y_0$. Em particular, para todo $x \in [c, b] \cap V_\delta(c)$, $f(x) < y_0$. Podemos supor $\delta < b - c$. Nesse caso, $f(x) < y_0$, para todo o $x \in [c, c + \delta[$. Consequentemente, os pontos do intervalo $[c, c + \delta[$ são pontos do conjunto X , o que contraria o facto de c ser um majorante (o supremo) de X . A contradição resulta de supor que se tem $f(c) < y_0$.

Se $f(c) > y_0$, então $c > a$ porque $f(a) < y_0$. Uma vez que f é contínua em c e $f(c) > y_0$, existe $\delta > 0$ tal que, para todo o $x \in [a, b] \cap V_\delta(c)$, $f(x) > y_0$. Em particular, para todo $x \in]c - \delta, c]$, $f(x) > y_0$. Deste modo, c não seria o supremo de X , o que contraria a hipótese. Esta contradição resultou de se ter suposto que $f(c) > y_0$.

Portanto, podemos concluir que $f(c) = y_0$ (e também, evidentemente, que $a < c < b$), como se pretendia. \square

¹Todo o subconjunto não vazio de \mathbb{R} , limitado superiormente, tem supremo.

Resulta do Teorema de Bolzano (Teorema 1.1) o corolário seguinte que permite localizar zeros de certas funções.

Corolário 1.1 *Se f é contínua em $[a, b]$ e $f(a) \times f(b) < 0$, então existe $c \in]a, b[$ tal que $f(c) = 0$.*

Prova: A hipótese garante que $f(a)$ e $f(b)$ têm sinais contrários, pelo que $f(a) \neq f(b)$ e ou $f(a) < 0 < f(b)$ ou $f(b) < 0 < f(a)$. Estamos então nas condições de aplicação do Teorema de Bolzano e podemos concluir que existe $c \in]a, b[$ tal que $f(c) = 0$, como pretendíamos. \square

Exemplo 1.1 *Consideremos a função f definida, em $\left[\frac{1}{\pi}, 1\right]$, por $f(x) = x \sin \frac{1}{x} + \ln(x^2)$.*

Uma vez que:

- a função f é contínua no intervalo $\left[\frac{1}{\pi}, 1\right]$;
- $f\left(\frac{1}{\pi}\right) = \frac{1}{\pi} \sin \pi + \ln \left(\frac{1}{\pi}\right)^2 = -2 \ln \pi < 0$;
- $f(1) = \sin 1 > 0$;

tem-se que $f\left(\frac{1}{\pi}\right) \times f(1) < 0$ e o corolário 1.1 permite concluir que existe pelo menos um $x_0 \in \left]\frac{1}{\pi}, 1\right[$ tal que $f(x_0) = 0$.

Exemplo 1.2 *Seja g a função definida, em $[0, 2\pi]$, por $g(x) = -x^2 \cos(2x) - \sin\left(2x - \frac{\pi}{3}\right)$.*

Uma vez que:

- a função g é contínua no intervalo $[0, 2\pi]$;
- $g(0) = -0 \times \cos 0 - \sin\left(0 - \frac{\pi}{3}\right) = \frac{\sqrt{3}}{2} > 0$;
- $g(2\pi) = -(2\pi)^2 \times \cos(2 \times 2\pi) - \sin\left(2 \times 2\pi - \frac{\pi}{3}\right) = \frac{\sqrt{3} - 8\pi^2}{2} < 0$;

tem-se que $g(0) \times g(2\pi) < 0$ e o corolário 1.1 permite concluir que a equação $g(x) = 0$ tem pelo menos uma solução no intervalo $]0, 2\pi[$.

O Problema (1.1) terá solução se existir um

$$x^* \in \mathcal{A} \text{ tal que } f(x^*) \leq f(x), \quad \forall x \in \mathcal{A}.$$

Portanto, tal problema pode não ter solução, como mostram as funções f e j representadas graficamente na Figura 1.4. A função g , nessa figura, tem máximo e mínimo absolutos. Ao contrário das outras três funções, está definida num intervalo fechado e é contínua nesse intervalo. Verificadas estas condições, o Teorema de Weierstrass (Teorema 1.2) garante a existência de extremos absolutos. Por conseguinte, sob condições de continuidade da função objetivo e compacticidade da região admissível, o Problema (1.1) tem solução. No entanto, como mostra a função h , uma função pode não ser contínua num intervalo fechado e ter mínimo absoluto nesse intervalo (o mínimo absoluto da função h é $h(7) = -1$).

Teorema 1.2 (Teorema de Weierstrass) *Sejam $a, b \in \mathbb{R}$ tais que $a < b$ e $f : [a, b] \rightarrow \mathbb{R}$ uma função. Se f é contínua em $[a, b]$, então f atinge em $[a, b]$ o máximo e o mínimo absolutos, isto é, existem $x_1, x_2 \in [a, b]$ tais que*

$$f(x_1) \leq f(x) \leq f(x_2)$$

para todo o $x \in [a, b]$, ou seja, o conjunto $f([a, b])$ admite máximo e mínimo.

Prova: Em primeiro lugar vamos demonstrar que se f é contínua em $[a, b]$, então o conjunto $f([a, b])$ é limitado e, portanto, existem $s, i \in \mathbb{R}$ tais que $s = \sup f([a, b])$ e $i = \inf f([a, b])$. Em seguida, demonstraremos que existe $x_1 \in [a, b]$ tal que $f(x_1) = s$ e que existe $x_2 \in [a, b]$ tal que $f(x_2) = i$, ou seja, que s é o máximo de $f([a, b])$ e que i é o mínimo de $f([a, b])$. Vamos provar a existência de x_1 nas condições indicadas. Para x_2 procede-se de modo análogo. Por definição de supremo de $f([a, b])$, temos $f(x) \leq s$, para todo o $x \in [a, b]$.

Seja $\delta > 0$. Como $s - \delta < s$ e f é contínua em $[a, b]$, o Teorema de Bolzano garante que existe $y \in [a, b]$ tal que $f(y) > s - \delta$. Admitamos, por redução ao absurdo, que, para todo o $x \in [a, b]$, $f(x) < s$. Consideremos a função

$$\begin{aligned} g : [a, b] &\longrightarrow \mathbb{R} \\ x &\longmapsto \frac{1}{s - f(x)}. \end{aligned}$$

Então g é uma função contínua em $[a, b]$. Consequentemente g é limitada em $[a, b]$. Mas, para todo o $\delta > 0$, existe $y \in [a, b]$ tal que $f(y) > s - \delta$, ou seja, $s - f(y) < \delta$ donde resulta que $g(y) > \frac{1}{\delta}$. Então g não é limitada superiormente, o que é falso, uma vez que g é uma função contínua num intervalo fechado e limitado. A contradição resulta de supor que, para todo o $x \in [a, b]$, $f(x) < s$, donde resulta o pretendido. \square

A otimização contínua dedica-se à minimização de funções contínuas definidas em compactos. Para irmos ao encontro do propósito deste trabalho e, sobretudo, para termos uma

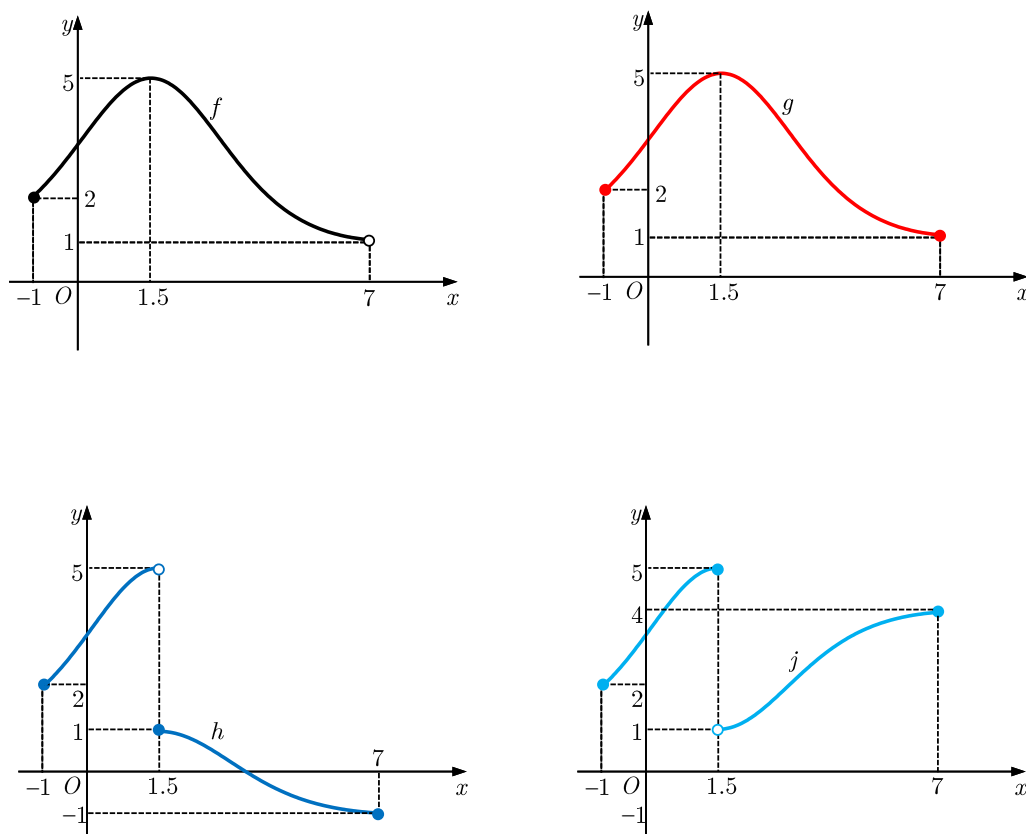


Figura 1.4: f é contínua e não tem mínimo absoluto. g é contínua e tem mínimo e máximo absolutos. h não é contínua e tem mínimo absoluto. j não é contínua, tem máximo absoluto e não tem mínimo absoluto.

garantia (suficiente) de solução para o Problema (1.1), vamos reformulá-lo enunciando-o do seguinte modo:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in \mathcal{A}, \quad \text{com } \mathcal{A} = [a, b] \text{ e } f \text{ contínua.} \end{array} \quad (1.2)$$

1.3 Otimização de Funções Convexas

Na prática, é frequente uma função ter vários extremos locais e, conseqüentemente, gera-se a incerteza sobre se o ponto extremo localizado por um algoritmo de otimização é o absoluto ou não. Nas funções convexas, qualquer minimizante local é minimizante global. Por oposição, nas funções côncavas qualquer maximizante local é maximizante global. Portanto, nestes casos, a otimização pode ser conseguida calculando extremantes locais. Nesta secção, estudaremos algumas propriedades dessas funções.

Suponhamos que temos uma função f definida no intervalo $[-1, 7]$. Tomemos como exemplo a função f da Figura 1.5 e consideremos alguns segmentos de reta cujos extremos são pontos do gráfico de f . Note-se que esses segmentos, como a figura documenta, estão acima do gráfico de f ou sobre este. Como, de facto, nesta função esta característica se verifica para quaisquer dois pontos no intervalo $[-1, 7]$, diz-se que, nesse intervalo, f é convexa. Uma vez que, no intervalo $[1, 5]$, os segmentos de reta que unem dois pontos do gráfico da função f situam-se sobre o gráfico, f não é estritamente convexa.

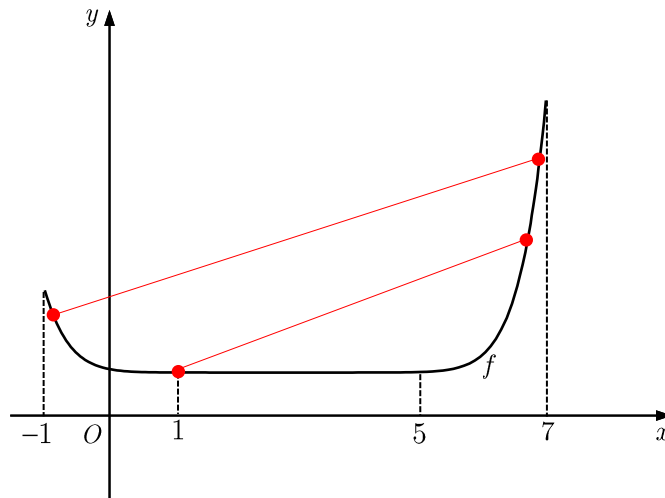


Figura 1.5: f é convexa no intervalo $[-1, 7]$.

Definição 1.3 (função convexa) *Seja I um intervalo de números reais.*

Uma função $f : I \rightarrow \mathbb{R}$ é convexa em I se, $\forall x_1, x_2 \in I, f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2), \forall \alpha \in [0, 1]$.

No caso de a desigualdade anterior se verificar em sentido estrito, para $x_1 \neq x_2$ e para todo $\alpha \in]0, 1[$, a função f diz-se estritamente convexa em I .

Como sugere a Figura 1.6, se numa função f , num dado intervalo, qualquer interpolação linear entre dois pontos majora o valor da função, então f é convexa nesse intervalo. Para $\alpha \in [0, 1]$, $y = \alpha f(x_1) + (1 - \alpha)f(x_2)$ é uma equação do segmento de reta $[AB]$.

Considerando o exemplo da Figura 1.5, a função $-f$ é côncava em $[-1, 7]$, em virtude de f ser convexa nesse intervalo.

Definição 1.4 (função côncava) *Seja I um intervalo de números reais. Uma função $f : I \rightarrow \mathbb{R}$ é côncava em I se, $\forall x_1, x_2 \in I, f(\alpha x_1 + (1 - \alpha)x_2) \geq \alpha f(x_1) + (1 - \alpha)f(x_2), \forall \alpha \in [0, 1]$.*

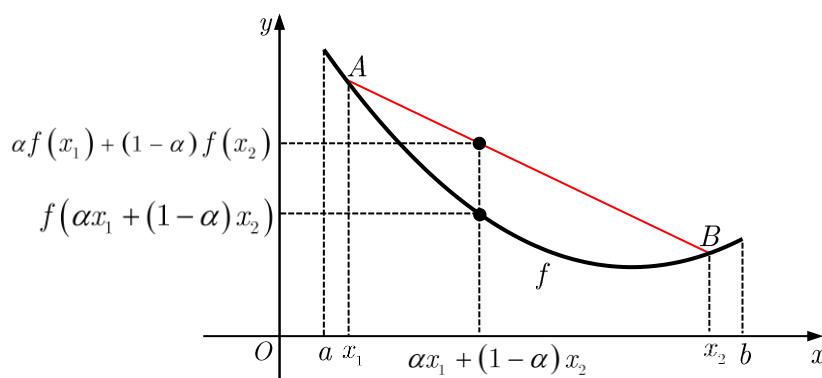


Figura 1.6: Ilustração da definição de função convexa.

No caso de a desigualdade anterior apenas se verificar em sentido estrito, para $x_1 \neq x_2$ e para todo $\alpha \in]0, 1[$, a função f diz-se estritamente côncava em I .

Na Figura 1.7 está representada uma função f que não é convexa nem côncava no intervalo $[-1, 5]$. Nesse intervalo, há uma parte do segmento de reta $[AB]$ abaixo do gráfico de f e outra acima. No entanto, no intervalo $[-1, 2]$ a função é côncava e no intervalo $[2, 5]$ a função é convexa.

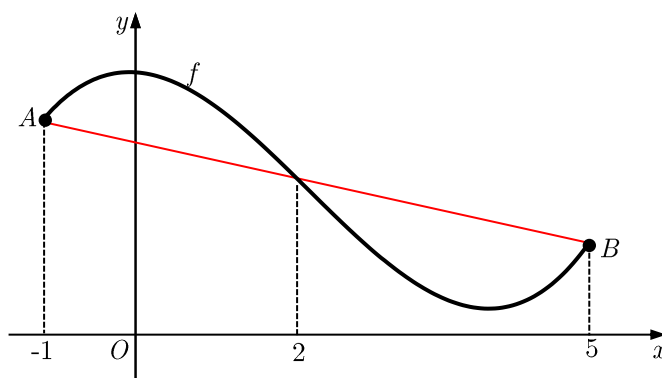


Figura 1.7: Função nem convexa nem côncava em $[-1, 5]$.

Em geral, não se consegue saber se um minimizante relativo é minimizante global ou não. No entanto, sob condições de convexidade da função objetivo temos a garantia de que os minimizantes relativos são globais. Este facto, serve para fundamentar a importância das funções convexas em otimização. Analisaremos a seguir algumas condições suficientes de otimalidade global.

Teorema 1.3 (unicidade do minimizante global) *Se $f : [a, b] \rightarrow \mathbb{R}$ é uma função estritamente convexa em $[a, b]$, então atinge o mínimo global num único ponto em $[a, b]$.*

Prova: Sejam $x_1^*, x_2^* \in [a, b]$ dois minimizantes globais de f tais que $x_1^* \neq x_2^*$ e $\alpha \in]0, 1[$. Da convexidade de $[a, b]$, resulta que $\alpha x_1^* + (1 - \alpha)x_2^* \in [a, b]$. Logo,

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) \geq f(x_1^*). \quad (1.3)$$

Dado que f é estritamente convexa e $f(x_1^*) = f(x_2^*)$,

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) < \alpha f(x_1^*) + (1 - \alpha)f(x_2^*)$$

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) < \alpha f(x_1^*) + (1 - \alpha)f(x_1^*)$$

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) < \alpha f(x_1^*) + f(x_1^*) - \alpha f(x_1^*)$$

$$f(\alpha x_1^* + (1 - \alpha)x_2^*) < f(x_1^*). \quad (1.4)$$

Das desigualdades (1.3) e (1.4), resulta uma contradição, a qual decorre de se ter suposto que $x_1^* \neq x_2^*$. Portanto, $x_1^* = x_2^*$ e o minimizante global de f , em $[a, b]$, é único. \square

Teorema 1.4 (condição suficiente de otimalidade global — funções convexas) *Se $f : [a, b] \rightarrow \mathbb{R}$ é uma função convexa em $[a, b]$, então qualquer minimizante relativo de f é minimizante global em $[a, b]$.*

Prova: Seja $x^* \in [a, b]$ um minimizante relativo de f . Então, existe $\epsilon > 0$ tal que, para todo $x \in [a, b]$ e $|x - x^*| < \epsilon$, $f(x^*) \leq f(x)$.

Pretendemos mostrar que para qualquer $y \in [a, b]$, $f(x^*) \leq f(y)$. Vamos, então, considerar $\alpha \in]0, 1[$ suficientemente pequeno, de modo que $x^* + \alpha(y - x^*) \in [a, b]$ e $|x^* + \alpha(y - x^*) - x^*| < \epsilon$. Consequentemente, e atendendo a que f é uma função convexa em $[a, b]$, tem-se

$$f(x^*) \leq f(x^* + \alpha(y - x^*)) = f(\alpha y + (1 - \alpha)x^*) \leq \alpha f(y) + (1 - \alpha)f(x^*).$$

Logo, se na desigualdade $f(x^*) \leq \alpha f(y) + (1 - \alpha)f(x^*)$ subtrairmos $f(x^*)$ aos dois membros

e, em seguida, dividirmos os dois membros por α , obtém-se

$$\frac{f(x^*) - f(x^*)}{\alpha} \leq \frac{\alpha f(y) + (1 - \alpha)f(x^*) - f(x^*)}{\alpha}$$

$$0 \leq \frac{\alpha f(y) + (1 - \alpha - 1)f(x^*)}{\alpha}$$

$$0 \leq f(y) - f(x^*)$$

$$f(x^*) \leq f(y).$$

Portanto, x^* é um minimizante global de f em $[a, b]$.

□

O teorema seguinte, que é intuitivamente plausível, é muitas vezes útil.

Teorema 1.5 (localização de minimizantes — funções estritamente côncavas) *Se $f : [a, b] \rightarrow \mathbb{R}$ é uma função estritamente côncava em $[a, b]$, então atinge o mínimo relativo (e global) em pelo menos um dos extremos do intervalo $[a, b]$.*

Prova: Seja $x \in]a, b[$ tal que, para $\alpha \in]0, 1[$, $x = \alpha a + (1 - \alpha)b$.

Uma vez que f é estritamente côncava em $[a, b]$,

$$f(x) > \alpha f(a) + (1 - \alpha)f(b). \quad (1.5)$$

Se:

- $f(a) > f(b)$, da Desigualdade (1.5), resulta

$$f(x) > \alpha f(b) + (1 - \alpha)f(b) \Leftrightarrow f(x) > f(b); \quad (1.6)$$

- $f(a) < f(b)$, da Desigualdade (1.5), resulta

$$f(x) > \alpha f(a) + (1 - \alpha)f(a) \Leftrightarrow f(x) > f(a); \quad (1.7)$$

- $f(a) = f(b)$, da Desigualdade (1.5), resulta

$$f(x) > \alpha f(a) + (1 - \alpha)f(a) \Leftrightarrow f(x) > f(a). \quad (1.8)$$

Das desigualdades (1.6), (1.7) e (1.8), conclui-se que o mínimo de f é $f(a)$ ou $f(b)$. □

As funções convexas diferenciáveis e com derivada contínua podem ser caracterizadas de forma alternativa à Definição 1.3. De acordo com o resultado seguinte, as funções convexas num intervalo são aquelas cujo gráfico, em cada ponto, está acima ou sobre a reta tangente ao gráfico nesse ponto.

Teorema 1.6 (condição necessária e suficiente de convexidade) *Seja f é uma função de classe \mathcal{C}^1 num intervalo I . Então,*

- (i) *f é convexa em I se, e só se, $f(x) \geq f(x_0) + f'(x_0)(x - x_0), \forall x, x_0 \in I$,*
- (ii) *f é estritamente convexa em I se, e só se, $f(x) > f(x_0) + f'(x_0)(x - x_0), \forall x, x_0 \in I$ e $x \neq x_0$.*

Prova: Para provar (i), suponha-se, primeiro, que f é convexa em I . Então, $\forall \alpha \in]0, 1]$ e $\forall x, x_0 \in I$

$$f(\alpha x + (1 - \alpha)x_0) < \alpha f(x) + (1 - \alpha)f(x_0)$$

$$f(\alpha x + x_0 - \alpha x_0) < \alpha f(x) + f(x_0) - \alpha f(x_0)$$

$$f(x_0 + \alpha(x - x_0)) < \alpha(f(x) - f(x_0)) + f(x_0)$$

$$\frac{f(x_0 + \alpha(x - x_0)) - f(x_0)}{\alpha} < f(x) - f(x_0). \quad (1.9)$$

Da Desigualdade (1.9) e atendendo que $\alpha(x - x_0) \rightarrow 0$ quando $\alpha \rightarrow 0$, resulta

$$\lim_{\alpha \rightarrow 0} \frac{f(x_0 + \alpha(x - x_0)) - f(x_0)}{\alpha} \leq \lim_{\alpha \rightarrow 0} (f(x) - f(x_0))$$

$$(x - x_0) \lim_{\alpha \rightarrow 0} \frac{f(x_0 + \alpha(x - x_0)) - f(x_0)}{\alpha(x - x_0)} \leq f(x) - f(x_0)$$

$$(x - x_0)f'(x_0) \leq f(x) - f(x_0)$$

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0).$$

O que prova que se f é convexa em I , então $f(x) \geq f(x_0) + f'(x_0)(x - x_0), \forall x, x_0 \in I$.

Suponha-se, agora, que $f(y) \geq f(z) + f'(z)(y - z), \forall y, z \in I$. Então, dados $x, x_1, x_2 \in I$ tem-se

$$f(x_1) \geq f(x) + f'(x)(x_1 - x) \quad (1.10)$$

e

$$f(x_2) \geq f(x) + f'(x)(x_2 - x). \quad (1.11)$$

Para $\alpha \in [0, 1]$, multiplicando a Desigualdade (1.10) por α e a Desigualdade (1.11) por $1 - \alpha$, obtém-se, respetivamente,

$$\alpha f(x_1) \geq \alpha f(x) + \alpha f'(x)(x_1 - x) \quad (1.12)$$

e

$$(1 - \alpha)f(x_2) \geq (1 - \alpha)f(x) + (1 - \alpha)f'(x)(x_2 - x). \quad (1.13)$$

Somando, ordenadamente, as desigualdades (1.12) e (1.13) e tomando $x = \alpha x_1 + (1 - \alpha)x_2$, obtém-se

$$\alpha f(x_1) + (1 - \alpha)f(x_2) \geq \alpha f(x) + \alpha f'(x)(x_1 - x) + (1 - \alpha)f(x) + (1 - \alpha)f'(x)(x_2 - x)$$

$$\alpha f(x_1) + (1 - \alpha)f(x_2) \geq \alpha f(x) + f(x) - \alpha f(x) + f'(x)[(1 - \alpha)(x_2 - x) + \alpha(x_1 - x)]$$

$$\alpha f(x_1) + (1 - \alpha)f(x_2) \geq f(x) + f'(x)(x_2 - x - \alpha x_2 + \alpha x + \alpha x_1 - \alpha x)$$

$$\alpha f(x_1) + (1 - \alpha)f(x_2) \geq f(\alpha x_1 + (1 - \alpha)x_2) + f'(x)(x_2 - \alpha x_1 - (1 - \alpha)x_2 - \alpha x_2 + \alpha x_1)$$

$$\alpha f(x_1) + (1 - \alpha)f(x_2) \geq f(\alpha x_1 + (1 - \alpha)x_2) + f'(\alpha x_1 + (1 - \alpha)x_2) \times 0$$

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2). \quad (1.14)$$

Dada a arbitrariedade na escolha de x_1 e x_2 , da Desigualdade (1.14) e atendendo à Definição 1.3, conclui-se que f é convexa em I .

A prova de (ii) será omitida por ser muito semelhante à apresentada para (i) e não acrescentar nada de novo relativamente a esta.

□

Admitamos que f é uma função convexa nas condições do Teorema 1.6, x e x^* são dois pontos de um intervalo I e $f'(x^*) = 0$. O Teorema 1.6 garante que $f(x) \geq f(x^*) + f'(x^*)(x - x^*)$. Como $f'(x^*) = 0$, a desigualdade anterior é equivalente a $f(x) \geq f(x^*)$. Dado que x é um elemento qualquer de I , então x^* é um minimizante global de f em I . Esta discussão, prova o seguinte importante e útil resultado.

Corolário 1.2 (ponto crítico de uma função convexa) *Se f é uma função convexa de classe \mathcal{C}^1 num intervalo I , então qualquer ponto crítico de f em I é um minimizante global de f .*

As definições de função convexa, de função estritamente convexa e as suas caracterizações descritas no Teorema 1.6 são ferramentas úteis para deduzir propriedades importantes desta classe de funções. No entanto, não são muito úteis para reconhecer funções convexas ou estritamente convexas em casos concretos. O teorema seguinte fornece um meio eficaz para reconhecer funções convexas ou estritamente convexas, nos casos em que a função é de classe \mathcal{C}^2 num intervalo aberto I .

Teorema 1.7 (condição necessária e suficiente de convexidade) *Seja f uma função de classe \mathcal{C}^2 $]a, b[$.*

(i) f é convexa em $]a, b[$ se, e só se, $f''(x) \geq 0, \forall x \in]a, b[$.

(ii) f é estritamente convexa em $]a, b[$ se, e só se, $f''(x) > 0, \forall x \in]a, b[$.

Prova: Vamos provar (i). Seja f uma função convexa em $]a, b[$. Para todo $x, x_0 \in]a, b[$

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0), \quad (1.15)$$

de acordo com o Teorema 1.6, e

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x - x_0)^2}{2!} \left(f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2} \right), \quad (1.16)$$

atendendo à fórmula de Taylor, onde $R_2(x)$ é o resto de ordem 2 e $\lim_{x \rightarrow x_0} \frac{R_2(x)}{(x - x_0)^2} = 0$.

Uma vez que a Igualdade (1.16) é equivalente a

$$f(x) - f(x_0) - f'(x_0)(x - x_0) = \frac{(x - x_0)^2}{2!} \left(f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2} \right)$$

e atendendo à Igualdade (1.15), conclui-se que

$$\frac{(x - x_0)^2}{2!} \left(f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2} \right) \geq 0. \quad (1.17)$$

Como o sinal de $f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2}$, numa vizinhança de x_0 , é o de $f''(x_0)$, dado que

$\lim_{x \rightarrow x_0} \frac{R_2(x)}{(x - x_0)^2} = 0$, da Desigualdade (1.17) conclui-se que $f''(x_0) \geq 0$.

Suponha-se, agora, que para um certo x_0 escolhido arbitrariamente em $]a, b[$, $f''(x_0) \geq 0$. Vamos mostrar que f é convexa em $]a, b[$.

Aplicando a fórmula de Taylor tem-se, para qualquer $x \in]a, b[$,

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x - x_0)^2}{2!} \left(f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2} \right). \quad (1.18)$$

Como o sinal de $f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2}$, numa vizinhança de x_0 , é o de $f''(x_0)$, dado que

$\lim_{x \rightarrow x_0} \frac{R_2(x)}{(x - x_0)^2} = 0$,

$$\frac{(x - x_0)^2}{2!} \left(f''(x_0) + \frac{2!R_2(x)}{(x - x_0)^2} \right) \geq 0$$

então, da Igualdade (1.18), resulta que

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0).$$

De acordo com o Teorema 1.6, f é convexa em $]a, b[$.

A demonstração de (ii) é muito semelhante à de (i), razão pela qual não será aqui apresentada. □

Note-se que na proposição (ii) não se pode fechar o intervalo. Veja-se, por exemplo, a função definida por $f(x) = x^3$. Tem-se $f''(x) = 6x$. Portanto, $f''(0) = 0$ mas a função f é estritamente convexa em $[0, 1]$.

1.4 Otimização Clássica

Os métodos clássicos de otimização são úteis para determinar extremos de funções contínuas e diferenciáveis. Estes métodos são analíticos e fazem uso de técnicas do cálculo diferencial na localização de pontos ótimos. Uma vez que muitos problemas práticos envolvem funções objetivo que não são contínuas ou diferenciáveis, as técnicas clássicas de otimização têm um alcance limitado. No entanto, o estudo desses métodos são o fundamento do desenvolvimento de grande parte dos métodos numéricos de otimização, alguns dos quais abordaremos nos capítulos seguintes. Nesta secção, estudaremos as condições necessárias e as condições suficientes para a determinação de soluções ótimas e discutiremos a solução de alguns problemas de otimização.

O conhecido Teorema de Fermat, que a seguir se enuncia, dá-nos um meio para encontrarmos pontos candidatos a extremos, sob condições de diferenciabilidade da função objetivo. Esses pontos, são os zeros da primeira derivada da função objetivo, que designaremos por *pontos de estacionaridade* ou *pontos críticos*.

Teorema 1.8 (Teorema de Fermat) *Seja $f :]a, b[\rightarrow \mathbb{R}$ uma função. Se f admite um extremo relativo em $c \in]a, b[$ e é diferenciável em c , então $f'(c) = 0$.*

Prova:

- Suponha-se que f tem um máximo relativo em $c \in]a, b[$. Então, existe um $\epsilon > 0$ tal que

$$\forall x \in]c - \epsilon, c + \epsilon[, f(x) \leq f(c).$$

Portanto, como $\forall x \in]c - \epsilon, c[, x - c < 0$ e $\forall x \in]c, c + \epsilon[, x - c > 0$,

$$f'_e(c) = \lim_{x \rightarrow c^-} \frac{f(x) - f(c)}{x - c} \geq 0 \text{ e } f'_d(c) = \lim_{x \rightarrow c^+} \frac{f(x) - f(c)}{x - c} \leq 0.$$

Como f é diferenciável em c , $f'(c) = f'_e(c) = f'_d(c) = 0$.

- Suponha-se que f tem um mínimo relativo em $c \in]a, b[$. Então, existe um $\epsilon > 0$ tal que

$$\forall x \in]c - \epsilon, c + \epsilon[, f(x) \geq f(c).$$

Portanto, como $\forall x \in]c - \epsilon, c[, x - c < 0$ e $\forall x \in]c, c + \epsilon[, x - c > 0$,

$$f'_e(c) = \lim_{x \rightarrow c^-} \frac{f(x) - f(c)}{x - c} \geq 0 \text{ e } f'_d(c) = \lim_{x \rightarrow c^+} \frac{f(x) - f(c)}{x - c} \leq 0.$$

Como f é diferenciável em c , $f'(c) = f'_e(c) = f'_d(c) = 0$.

□

O recíproco do Teorema 1.8 (Teorema de Fermat) é falso. De facto, uma função pode não ter extremo num ponto em que a derivada se anula. A título de exemplo, veja-se a função f representada graficamente na Figura 1.8. Nessa função, $f'(2) = 0$, todavia $f(2)$ não é extremo, é um ponto de inflexão do gráfico da função f .

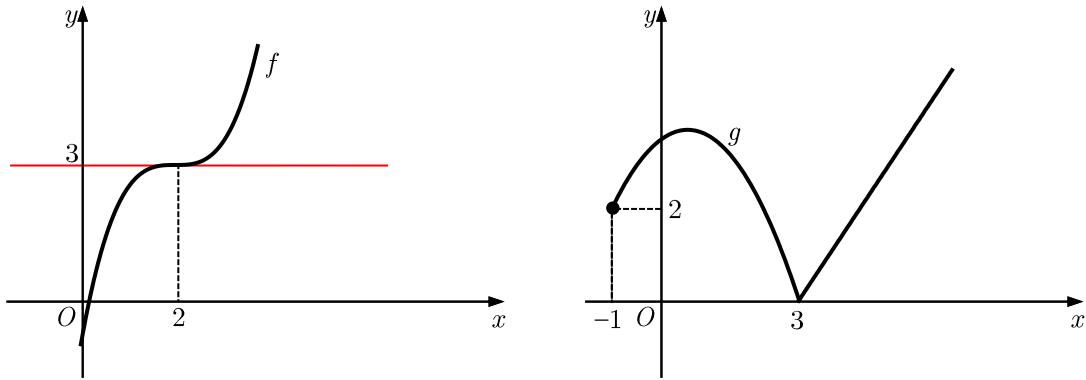


Figura 1.8: $f'(2) = 0$, mas $f(2)$ não é extremo. $g(3) = 0$ é mínimo absoluto, porém $g'(3)$ não existe.

O Teorema de Fermat não garante que os extremos só podem ocorrer em pontos de estacionaridade. Na verdade, o mínimo (os extremos em geral) pode ocorrer também em pontos em que a derivada não existe, sendo a função g representada na Figura 1.8 um exemplo ($g(3) = 0$ é mínimo absoluto, porém $g'(3)$ não existe, por $x = 3$ ser um ponto anguloso). Mais: nos pontos em que a derivada não existe, uma função pode ser descontínua e ter ou não mínimo. Na função h da Figura 1.9, $h'(3)$ e $h'(5)$ não existem, porque $x = 3$ e $x = 5$ são pontos de descontinuidade da função h . Em $x = 3$, h tem um mínimo relativo e em $x = 5$ não tem qualquer extremo. A função j é diferenciável e exemplifica o Teorema 1.8. Nos pontos $x = 2$ e $x = 6$, j tem um máximo e um mínimo, respetivamente, sendo nula a derivada (a reta tangente ao gráfico de j nesses pontos é uma reta de declive zero, uma reta horizontal).

Conhecidos os pontos de estacionaridade, importa saber se são ou não pontos de extremo. No caso de o serem, há todo interesse em classificá-los em pontos de máximo ou mínimo relativos. O resultado seguinte responde a esta questão.

Teorema 1.9 (condição suficiente) *Sejam $f : I \rightarrow \mathbb{R}$ uma função n vezes diferenciável num ponto x^* pertencente ao interior de I .*

Se $f'(x^) = f''(x^*) = \dots = f^{(n-1)}(x^*) = 0 \wedge f^{(n)}(x^*) \neq 0$, então $f(x^*)$:*

- (i) é um mínimo relativo de f se $f^{(n)}(x^*) > 0$ e n é par;*
- (ii) é um máximo relativo de f se $f^{(n)}(x^*) < 0$ e n é par;*
- (iii) não é extremo relativo se n é ímpar.*

Prova: Aplicando a fórmula de Taylor, de ordem $n - 1$, com resto de Peano, obtém-se

$$f(x) = f(x^*) + \dots + \frac{f^{(n-1)}(x^*)}{(n-1)!}(x - x^*)^{n-1} + \frac{(x - x^*)^n}{n!} \left(f^{(n)}(x^*) + n!\alpha(x) \right) \quad (1.19)$$

com $\alpha(x) = \frac{R_n(x)}{(x - x_0)^n}$ e $\lim_{x \rightarrow x^*} (\alpha(x)) = 0$.

Aplicando a hipótese à Igualdade (1.19), obtém-se

$$f(x) = f(x^*) + \frac{(x - x^*)^n}{n!} \left(f^{(n)}(x^*) + n!\alpha(x) \right)$$

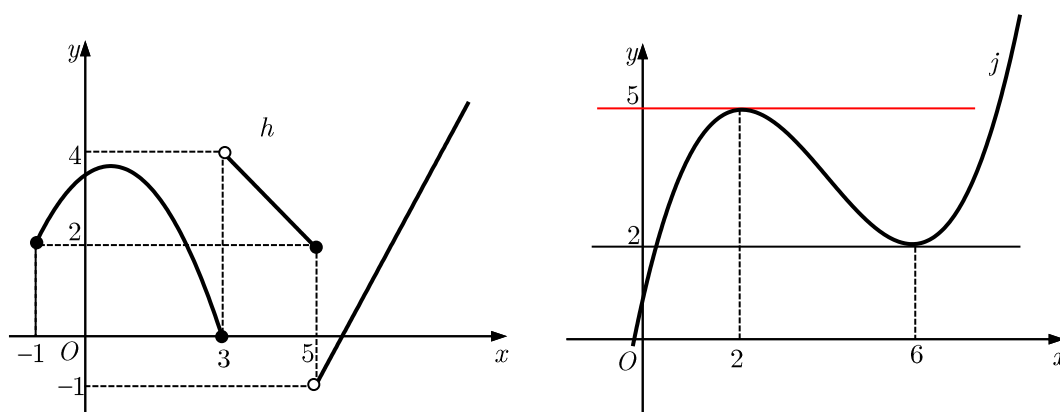


Figura 1.9: $h'(3)$ e $h'(5)$ não existem. Em $x = 3$ a função h tem um mínimo relativo, mas em $x = 5$ não há extremos. A função j diferenciável e tem derivada nula nos pontos extremantes.

ou seja,

$$f(x) - f(x^*) = \frac{(x - x^*)^n}{n!} \left(f^{(n)}(x^*) + n! \alpha(x) \right)$$

como $\lim_{x \rightarrow x^*} (n! \alpha(x)) = 0$, existe um $\epsilon > 0$ tal que para todo x pertencente a I e ao intervalo $]x^* - \epsilon, x^* + \epsilon[$, $f^{(n)}(x^*) + n! \alpha(x)$ tem o sinal de $f^{(n)}(x^*)$. Consequentemente, $f(x) - f(x^*)$ tem o sinal de $\frac{(x - x^*)^n}{n!} f^{(n)}(x^*)$. Se n for par, $\frac{(x - x^*)^n}{n!}$ é sempre positivo e o sinal de $f(x) - f(x^*)$ é igual ao sinal de $f^{(n)}(x^*)$. Se n for ímpar, $\frac{(x - x^*)^n}{n!}$ muda de sinal consoante $x > x^*$ ou $x < x^*$, o mesmo acontecendo a $f(x) - f(x^*)$. Portanto, cumpre-se (i), (ii) e (iii). \square

1.5 Aplicações

O método que analisamos neste capítulo, para encontrar valores extremos, tem aplicações práticas em muitas áreas do conhecimento humano. Um empresário quer minimizar os custos e maximizar o lucro. Um viajante quer minimizar o tempo de transporte. O princípio de Fermat em ótica afirma que a luz segue o caminho que leva menos tempo.

Nesta secção resolveremos alguns problemas de otimização de áreas, lucros, distâncias, tempos, ângulos e custos.

Na resolução de problemas práticos, o maior desafio é, muitas vezes, converter o texto do problema proposto para linguagem matemática, através da definição da função a maximizar ou minimizar e de eventuais restrições.

Problema 1.1 (Área mínima/Custo mínimo) *Uma fábrica pretende construir embalagens cilíndricas em alumínio para produtos de conserva, com capacidade igual a 1 litro.*

Quais devem ser as dimensões das embalagens para que o custo do alumínio seja mínimo?

Resolução:

Consideremos a Figura 1.10. Tendo em vista minimizar o custo do alumínio, vamos minimizar a área total de um cilindro. Tomando para unidade de comprimento o centímetro e sendo r a medida do raio da base e h a altura do cilindro, de acordo com a Figura 1.10, a área total do cilindro é dada por

$$A = 2\pi r h + 2\pi r^2.$$

Atendendo que a capacidade das embalagens é 1 litro, ou seja, 1000 cm^3 ,

$$\pi r^2 h = 1000 \Leftrightarrow h = \frac{1000}{\pi r^2}.$$

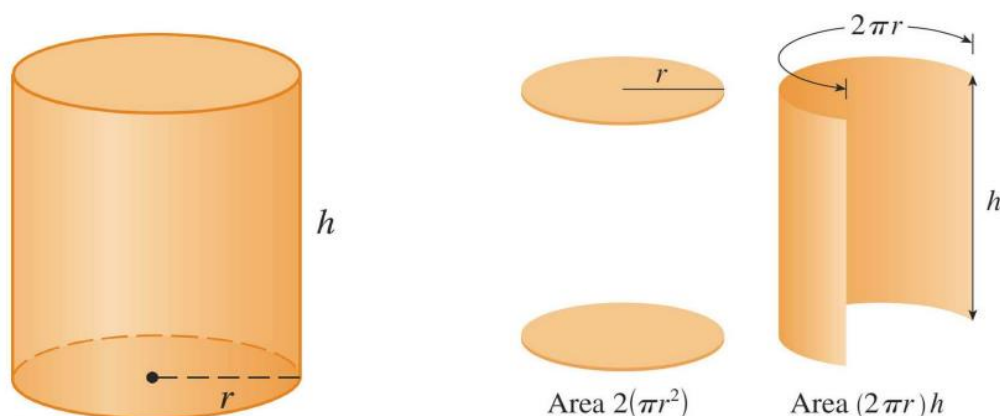


Figura 1.10: Decomposição de uma embalagem cilíndrica — Problema 1.1.

Donde resulta que a área é dada, em função de r , por

$$A(r) = 2\pi r \times \frac{1000}{\pi r^2} + 2\pi r^2 \Leftrightarrow A(r) = \frac{2000}{r} + 2\pi r^2$$

Portanto, temos que minimizar a função A , definida em $]0, +\infty[$, por

$$A(r) = \frac{2000}{r} + 2\pi r^2.$$

Para $r > 0$, a derivada de A é

$$A'(r) = 4\pi r - \frac{2000}{r^2}.$$

Uma vez que A é diferenciável em $]0, +\infty[$, caso existam, os extremos ocorrem em pontos tais que $A'(r) = 0$, de acordo com o Teorema 1.8 (pág. 20).

$$A'(r) = 0 \Leftrightarrow 4\pi r - \frac{2000}{r^2} = 0 \Leftrightarrow 4\pi r^3 - 2000 = 0 \Leftrightarrow r = \sqrt[3]{\frac{500}{\pi}}.$$

Vamos aplicar o Teorema 1.9 (pág. 22) para classificar o único ponto crítico da função A . A segunda derivada de A é, para $r > 0$,

$$A''(r) = 4\pi + \frac{4000}{r^3}.$$

Uma vez que

$$A''\left(\sqrt[3]{\frac{500}{\pi}}\right) = 12\pi > 0,$$

conclui-se que em $r = \sqrt[3]{\frac{500}{\pi}} = 5.419 \dots$ a função A tem um mínimo relativo.

A função A é estritamente convexa em $]0, +\infty[$, porque nesse intervalo $A''(r) > 0$, de acordo com o Teorema 1.7 (pág. 18). Logo, de acordo com o Teorema 1.4 (pág. 15), $r = \sqrt[3]{\frac{500}{\pi}}$ é um minimizante global.

O valor de h correspondente a $r = \sqrt[3]{\frac{500}{\pi}}$ é

$$h = \frac{1000}{\pi r^2} = \frac{1000}{\pi \left(\sqrt[3]{\frac{500}{\pi}} \right)^2} = 2 \sqrt[3]{\frac{500}{\pi}} = 10.838 \dots$$

Portanto, o custo do alumínio é mínimo se o raio das embalagens medir $\sqrt[3]{\frac{500}{\pi}} = 5.419 \dots$ cm e a sua altura for $2 \sqrt[3]{\frac{500}{\pi}} = 10.838 \dots$ cm.

Problema 1.2 (Caminho mais rápido) Um ciclista mora numa localidade que dista 20 quilómetros de uma estrada alcatroada retilínea. Diariamente, tem que se deslocar, de bicicleta, de sua casa para uma vila que se situa a 50 quilómetros do ponto, da estrada alcatroada, mais próximo da localidade onde mora.

Na estrada alcatroada o ciclista desloca-se a 25 Km/h e do local onde mora até à estrada alcatroada desloca-se a 15 Km/h, por uma estrada retilínea em terra batida.

Determine o ponto onde deve entrar na estrada alcatroada de modo a fazer o percurso no menor tempo possível.

Resolução:

Na Figura 1.11 está um esquema do problema proposto. O ponto L representa a localidade onde mora o ciclista, o ponto O representa o ponto da estrada alcatroada mais próximo de L , o ponto V representa a vila e o ponto P é o ponto da estrada alcatroada onde o ciclista deve entrar para fazer o percurso no menor tempo possível. x designa a distância, em Km, do ponto O ao ponto P .

Aplicando o Teorema de Pitágoras ao triângulo retângulo $[POL]$, resulta $\overline{PL} = \sqrt{400 + x^2}$. Considerando a relação $v = \frac{d}{t}$, onde v representa a velocidade, d o espaço e t o tempo, conclui-se que o ciclista demora $\frac{\sqrt{400 + x^2}}{15}$ a ir de L a P e $\frac{50 - x}{25}$ a ir de P a V . Portanto,

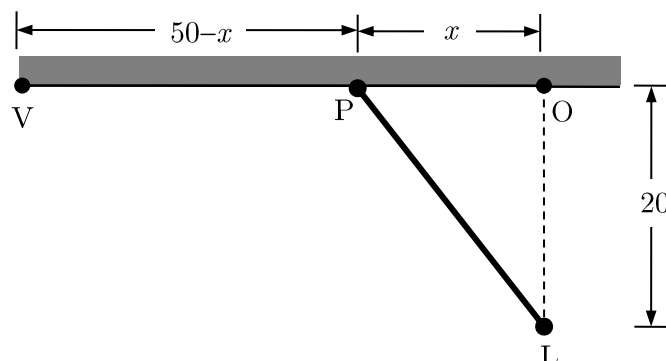


Figura 1.11: Esquematização do Problema 1.2.

o tempo total, em horas, é dado, em função de x , por

$$T(x) = \frac{\sqrt{400 + x^2}}{15} + \frac{50 - x}{25}.$$

Atendendo que x e $50 - x$ são distâncias, $x \geq 0 \wedge 50 - x \geq 0$, ou seja, $0 \leq x \leq 50$. Logo, o domínio da função T é $D_T = [0, 50]$.

Portanto, pretende-se minimizar a função T , definida em $[0, 50]$, por

$$T(x) = \frac{\sqrt{400 + x^2}}{15} + \frac{50 - x}{25}.$$

No intervalo $]0, 50[$, a derivada de T é

$$T'(x) = \frac{1}{15} \times \frac{2x}{2\sqrt{400 + x^2}} - \frac{1}{25} \Leftrightarrow T'(x) = \frac{5x - 3\sqrt{400 + x^2}}{75\sqrt{400 + x^2}}.$$

Uma vez que T é diferenciável em $]0, 50[$, T só pode ter extremos nos pontos em que $T'(x) = 0$, de acordo com o Teorema 1.8 (pág. 20). No intervalo $]0, 50[$ tem-se

$$\frac{5x - 3\sqrt{400 + x^2}}{75\sqrt{400 + x^2}} = 0 \Leftrightarrow 5x - 3\sqrt{400 + x^2} = 0 \Rightarrow 25x^2 = 9(400 + x^2) \Leftrightarrow 16x^2 = 3600 \Leftrightarrow x = 15.$$

Dado que $15 \in]0, 50[$ e

$$T'(15) = \frac{5 \times 15 - 3\sqrt{400 + 15^2}}{75\sqrt{400 + 15^2}} = 0,$$

então 15 é ponto crítico de T em $]0, 50[$

A segunda derivada de T é, para $0 < x < 50$,

$$T''(x) = \frac{80}{3\sqrt{(400 + x^2)^3}}.$$

Uma vez que $T''(x) > 0, \forall x \in \mathbb{R}$, a função T é convexa em \mathbb{R} , de acordo com o Teorema 1.7 (pág. 18), sendo, em particular convexa no intervalo $[0, 50]$. Logo, de acordo com o Corolário 1.2 (pág. 18), $x = 15$ é um minimizante global.

Por conseguinte, o ciclista, vindo de sua casa com destino à vila, deve entrar na estrada alcatroada no ponto situado a 15 Km do ponto, dessa estrada, mais próximo de sua casa.

Problema 1.3 (Melhor ângulo de visão) Numa sala de cinema, o ecrã está junto a uma parede que dista 50 metros da parede oposta, onde se encontra a máquina de projeção. O ecrã mede 20 metros de altura e está 10 metros acima do chão. A que distância da parede onde está o ecrã se deve posicionar um observador de modo que o seu ângulo de visão para tela de cinema seja o maior possível? Qual é a amplitude desse ângulo?

Resolução:

O problema será resolvido considerando o esquema apresentado na Figura 1.12. Nos triângulos retângulos $[BOP]$ e $[BOV]$, para $\alpha, \theta, \alpha + \theta \in \left]0, \frac{\pi}{2}\right[$, verifica-se

$$\begin{cases} \tan(\alpha) = \frac{10}{x} \\ \tan(\alpha + \theta) = \frac{30}{x} \end{cases} \Leftrightarrow \begin{cases} \alpha = \arctan\left(\frac{10}{x}\right) \\ \theta = \arctan\left(\frac{30}{x}\right) - \alpha \end{cases} \Leftrightarrow \begin{cases} \alpha = \arctan\left(\frac{10}{x}\right) \\ \theta = \arctan\left(\frac{30}{x}\right) - \arctan\left(\frac{10}{x}\right). \end{cases}$$

Seja $\theta = f(x)$. Como se pretende maximizar o ângulo de visão para o ecrã, a função objetivo é definida por

$$f(x) = \arctan\left(\frac{30}{x}\right) - \arctan\left(\frac{10}{x}\right)$$

Atendendo que x representa uma distância e o observador só consegue ver as imagens projetadas no ecrã se estiver afastado deste, $x > 0$. Como as paredes onde estão a máquina de projeção e o ecrã distam entre si 50 metros, $x \leq 50$. Logo, o domínio da função f é $D_f =]0, 50]$.

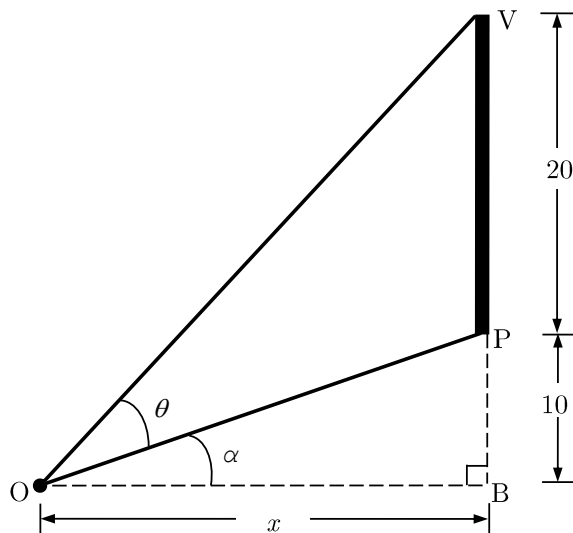


Figura 1.12: Esquematização do Problema 1.3.

Uma vez que f é diferenciável em $]0, 50[$, f só pode ter extremos nos pontos em que $f'(x) = 0$, de acordo com o Teorema 1.8 (pág. 20).

Vamos determinar o extremo relativo de f , no intervalo $]0, 50[$, usando a Symbolic Toolbox do MatLab. Para o efeito, foram usados os seguintes comandos:

```
syms x
f=atan(30/x)-atan(10/x)
D1f=diff(f,1)
D2f=diff(f,2)
S=solve(D1f)
Min=S(1)
D2Min=subs(D2f,Min)
fMin=subs(f,Min)
teta_max=fMin*180/pi
```

A compilação dos comandos anteriores produziu os seguintes resultados:

```
f=
atan(30/x)-atan(10/x)
D1f=
10/(x^2*(100/x^2+1))-30/(x^2*(900/x^2+1))
D2f=
```

```

2000/(x^5*(100/x^2+1)^2)-20/(x^3*(100/x^2+1))+60/(x^3*(900/x^2+1))
-54000/(x^5*(900/x^2+1)^2)
S=
10*3^(1/2)-10*3^(1/2)
Min=
10*3^(1/2)
D2Min=
-3^(1/2)/1200
fMin=
pi/6
teta_max=
30

```

Os resultados anteriores mostram que $f(10\sqrt{3}) = \frac{\pi}{6}$ é máximo relativo de f em $]0, 50[$.

Para $x \in]0, 10\sqrt{3}[$, $f'(x) > 0$ e, para $x \in]10\sqrt{3}, 50[$, $f'(x) < 0$. Como adicionalmente f é contínua no intervalo $]0, 50[$ e é diferenciável em $]0, 50[$, conclui-se que $x = 10\sqrt{3}$ é maximizante global de f em $]0, 50[$. Portanto, o maior ângulo de visão do observador é igual a 30° e ocorre quando este está a $10\sqrt{3} \approx 17.3$ metros da parede onde está colocada a tela de cinema.

Problema 1.4 (Lucro máximo) Pretende-se vender um determinado número de DVD. De acordo com a pesquisa de mercado que foi feita, se o preço unitário for de 1.50 €, será possível vender 5000 DVD e para cada redução de 0.10 € no preço, é possível vender mais 1000 DVD. Os custos fixos são de 2000.00 € e o custo de produção de um DVD é de 0.50 €. Determine o preço de cada DVD para que o lucro seja o maior possível, admitindo que todos os DVD são vendidos. Indique o lucro máximo e o número de DVD vendidos.

Resolução:

Seja $L(x)$ o lucro da venda de n DVD, quando o preço de cada DVD é x .

Como o lucro é a receita menos os custos e a receita é o número de DVD vendidos vezes o preço unitário, então

$$L(x) = nx - 2000 - 0.50n.$$

Uma vez que, por cada redução de 0.10 € no preço de cada DVD, é possível vender mais 1000 DVD, o par ordenado $(1.5, 5000)$ representa um ponto de uma reta de declive $\frac{1000}{-0.1}$. Uma equação dessa reta é

$$n - 5000 = \frac{1000}{-0.10}(x - 1.5).$$

Ou seja,

$$n = -10000x + 20000.$$

Assim sendo,

$$\begin{aligned} L(x) &= (-10000x + 20000) \times x - 2000 - 0.50 \times (-10000x + 20000) \Leftrightarrow \\ &\Leftrightarrow L(x) = -10000x^2 + 25000x - 12000. \end{aligned}$$

Como x representa o preço, $x \geq 0$. De $1.5 - x \geq 0$, resulta $x \leq 1.5$. Portanto, pretende-se maximizar a função L no intervalo $[0, 1.5]$. Para $x \in]0, 1.5[$,

$$L'(x) = -20000x + 25000$$

e

$$L''(x) = -20000.$$

Uma vez que L é diferenciável no intervalo $]0, 1.5[$, determinemos os seus pontos críticos nesse intervalo, pois apenas nesses pontos L pode ter extremos em $]0, 1.5[$, de acordo com o Teorema 1.8 (pág. 20). Nesse intervalo, tem-se

$$L'(x) = 0 \Leftrightarrow -20000x + 25000 = 0 \Leftrightarrow x = 1.25.$$

Como

$$L''(1.25) = -20000 < 0,$$

$x = 1.25$ é maximizante relativo de P em $]0, 1.5[$.

Atendendo que

$$L(0) = -12000, L(1.25) = 27625 \text{ e } L(1.5) = 27000,$$

conclui-se que $x = 1.25$ é maximizante global de L em $[0, 1.5]$. Nestas condições, são vendidos $n = -10000 \times 1.25 + 20000 = 7500$ DVD ao preço unitário de 1.25 € , obtendo-se um lucro de $27\,625.00 \text{ €}$.

Capítulo 2

Métodos Intervalares de Eliminação

Os métodos intervalares de eliminação são métodos iterativos que em cada iteração reduzem a amplitude do intervalo que contém o único minimizante x^* de f . Este propósito é alcançado usando valores da função objetivo f em pontos adequados. O processo iterativo termina quando for satisfeito o critério de paragem, o qual pode ser um certo número de iterações ou um intervalo contraído com amplitude inferior a um dado valor $\delta > 0$. Uma aproximação para x^* pode ser o valor médio do intervalo obtido na última iteração.

Uma vantagem destes métodos reside no facto de não exigirem a continuidade ou a diferenciabilidade de f . Apenas exigem que f seja unimodal, com mínimo, num intervalo $[a, b]$ (ver Definição 2.1 na Secção 2.1).

Para desenvolver este capítulo, teremos como referência os autores [1, Antoniou e Lou], [2, Biryukov], [5, Chong e Žak], [8, De Leon], [11, Sá Esteves], [12, Fernandes], [17, Hanson], [18, Higham], [20, Knight], [19, Kiusalaas], [21, Lima], [22, LoVetri], [23, Mathews e Fink] e [26, Rao].

2.1 Funções Unimodais

Sendo x^* um minimizante de uma função f estritamente convexa num intervalo $[a, b]$, podemos garantir que:

- (i) x^* é único em $[a, b]$;
- (ii) f é estritamente decrescente em $[a, x^*]$, desde que $a \neq x^*$;
- (iii) f é estritamente crescente em $[x^*, b]$, desde que $b \neq x^*$.

Há, no entanto, uma classe de funções — denominadas funções unimodais — que satisfazem as condições (i), (ii) e (iii), mas que não são estritamente convexas. É do estudo destas funções que nos ocuparemos nesta secção.

Definição 2.1 (função unimodal) *Seja f uma função definida num intervalo $[a, b]$.*

Diz-se que f é unimodal (com mínimo) em $[a, b]$ se atinge o seu mínimo global num único ponto $x^ \in]a, b[$ e para quaisquer $x_1, x_2 \in [a, b]$, se verificar:*

- (i) $f(x_2) < f(x_1)$ se $x_1 < x_2 < x^*$;
- (ii) $f(x_1) < f(x_2)$ se $x^* < x_1 < x_2$.

As funções representadas na Figura 2.1 são unimodais no intervalo $[1, 5]$, tendo qualquer uma delas mínimo no ponto x^* . Esse exemplo mostra que uma função unimodal pode não ser diferenciável — as funções g e h —, pode não ser contínua — a função g — e pode não ser convexa — a função h .

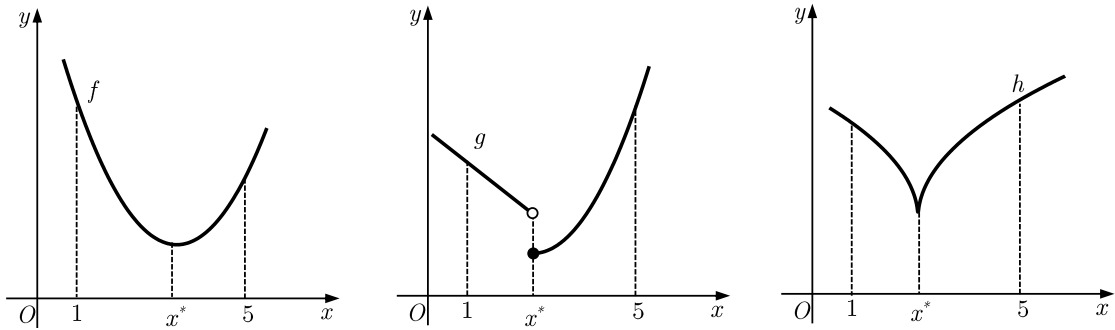


Figura 2.1: Exemplos de funções unimodais.

Seja f uma função unimodal (com mínimo) no intervalo $[a, b]$. Este intervalo é um intervalo de incerteza. Um intervalo de incerteza será aqui considerado qualquer intervalo que contenha o minimizante x^* de f , desde que não se conheça a localização exata de x^* .

Como já foi referido, x^* pode ser localizado reduzindo progressivamente o intervalo de incerteza até se obter um intervalo de amplitude suficientemente pequena. Nos métodos de busca, este propósito consegue-se usando valores de f em pontos adequados.

No caso de o valor de f ser conhecido num único ponto $x_1 \in]a, b[$, é igualmente provável x^* estar no intervalo $]a, x_1[$ ou no intervalo $]x_1, b[$, como mostra a Figura 2.2, onde o minimizante da função h pertence ao intervalo $]a, x_1[$ e o da função g pertence ao intervalo $]x_1, b[$. Por conseguinte, a informação disponível — valor da função objetivo num único ponto — não é suficiente para permitir a redução do intervalo de incerteza. No entanto, se o valor de f for conhecido em dois pontos $x_1, x_2 \in]a, b[$ ($x_1 < x_2$), é possível contrair o intervalo. Três casos podem ocorrer:

- (a) $f(x_1) < f(x_2)$. Como se ilustra na Figura 2.3, o minimizante x^* não pode estar no intervalo $[x_2, b]$. Portanto, no processo de procura de x^* podemos eliminar este intervalo, obtendo-se o intervalo de incerteza contraído $[a, x_2]$;

- (b) $f(x_1) > f(x_2)$. x^* não pode estar no intervalo $[a, x_1]$. Portanto, na procura de x^* podemos descartar este intervalo, resultando o intervalo de incerteza reduzido $[x_1, b]$, como se exemplifica na Figura 2.4;
- (c) $f(x_1) = f(x_2)$. O mínimo não pode estar no intervalo $[a, x_1]$ nem no intervalo $[x_2, b]$. Logo, estes intervalos podem ser eliminados no processo de procura de x^* (ver Figura 2.5). Pelo que resulta o intervalo de incerteza contraído $[x_1, x_2]$. No entanto, devido aos erros inerentes ao cálculo podemos não ter a garantia de que $f(x_1) = f(x_2)$, mas apenas que $f(x_1) \approx f(x_2)$. Por segurança, devemos escolher, para intervalo contraído, o intervalo $[a, x_2]$ ou o intervalo $[x_1, b]$. Note-se que $[x_1, x_2] \subset [a, x_2]$ e $[x_1, x_2] \subset [x_1, b]$.

Este processo pode ser repetido, desde que se tenha dois valores da função f no interior do intervalo de incerteza.

Observação 2.1 O caso (c) refere que a função objetivo pode ser calculada com erro, o que baralha os algoritmos tratados neste capítulo. Este assunto está estudado no livro *Optimum Seeking Methods*, de Douglass Wilde, editado em 1964 pela Prentice-Hall. \square

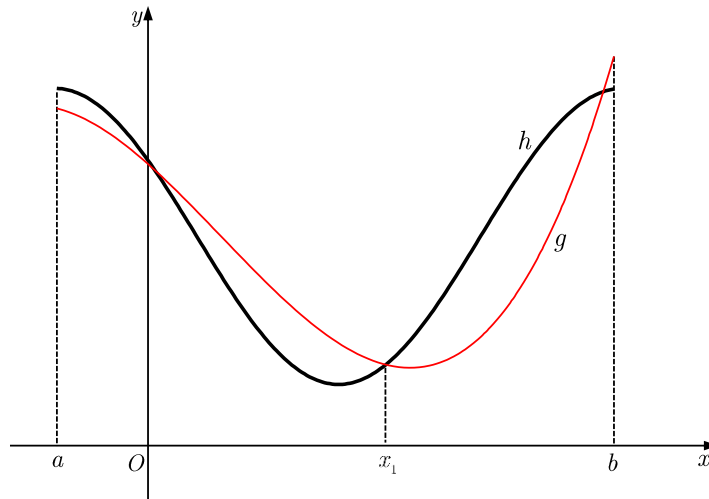


Figura 2.2: Impossibilidade de redução do intervalo de incerteza.

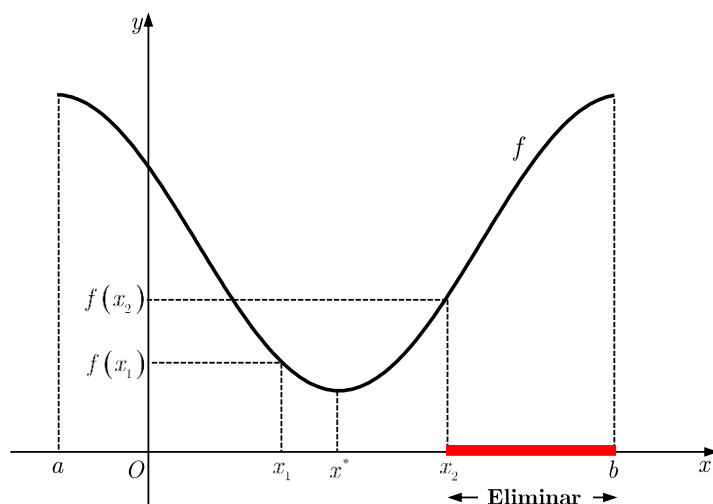


Figura 2.3: Redução do intervalo de incerteza quando $f(x_1) < f(x_2)$.

2.2 Método de Busca Dicotômica

Consideremos uma função unimodal f no intervalo $[a, b]$ e seja x^* o único minimizante de f em $[a, b]$.

Vamos descrever a primeira iteração. Para aplicar o método de busca dicotômica, são escolhidos dois pontos x_1 e x_2 suficientemente próximos do ponto médio do intervalo de incerteza. Portanto, x_1 e x_2 são colocados simetricamente, em relação ao ponto médio $\frac{b+a}{2}$ do intervalo de incerteza $[a, b]$, a uma distância $\varepsilon > 0$ deste, como se mostra na Figura 2.6 e na Figura 2.7. Logo,

$$x_1 = \frac{b+a}{2} - \varepsilon \text{ e } x_2 = \frac{b+a}{2} + \varepsilon.$$

O número positivo ε é escolhido de modo que seja suficientemente pequeno e $f(x_1)$ garantidamente diferente de $f(x_2)$. A função tem que variar alguma coisa entre x_1 e x_2 . Se, com a precisão da máquina, não houver variação, o processo não funciona. Na escolha de ε deve dar-se atenção à precisão da máquina de calcular ou computador (erro de representação) e ao erro gerado na avaliação da função objetivo. Adicionalmente, sendo δ a precisão do intervalo de incerteza, deve verificar-se

$$\varepsilon \ll \frac{b-a}{2} \text{ e } \varepsilon \ll \delta.$$

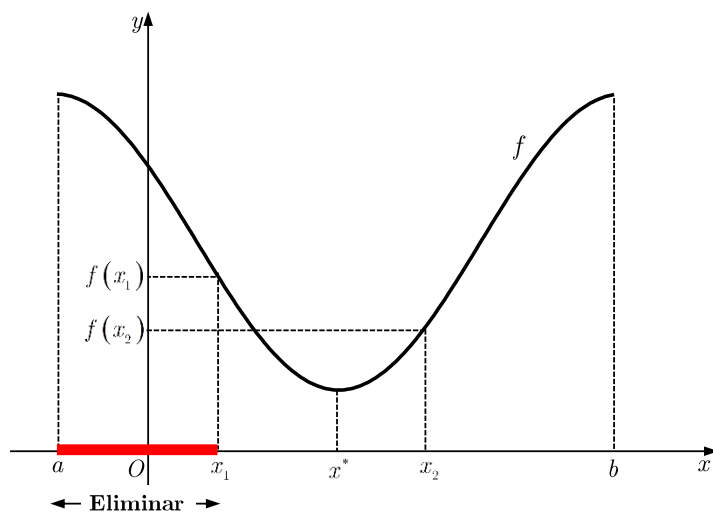


Figura 2.4: Redução do intervalo de incerteza quando $f(x_1) > f(x_2)$.

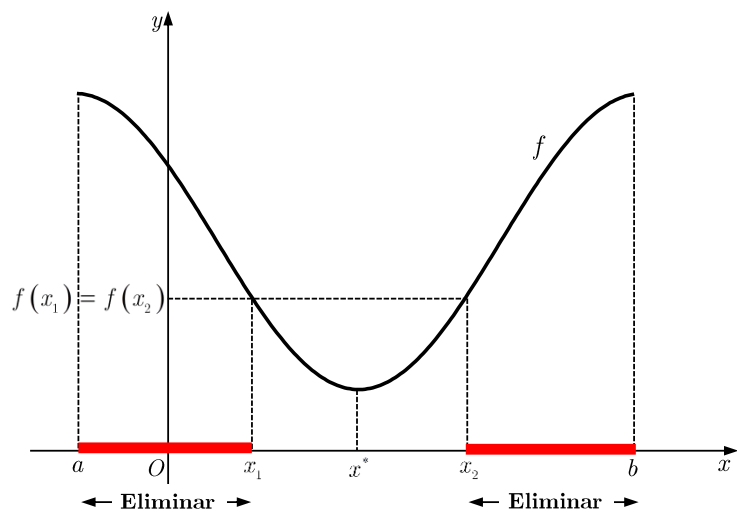
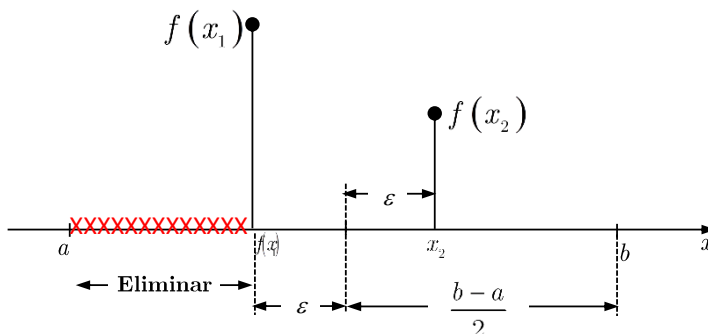
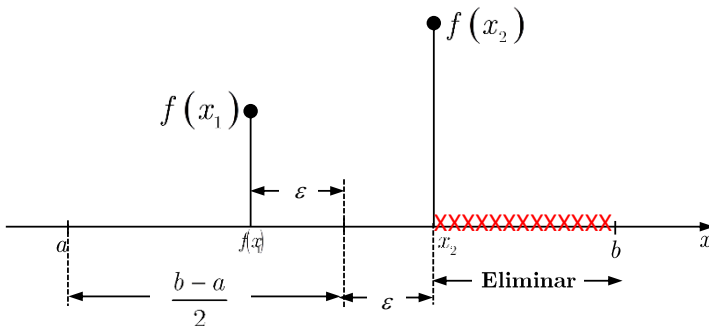


Figura 2.5: Redução do intervalo de incerteza quando $f(x_1) = f(x_2)$.

Uma vez que f é unimodal, se $f(x_1) < f(x_2)$, o novo intervalo de incerteza será $[a, x_2]$, cuja



$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in \mathcal{A}, \quad \text{com } \mathcal{A} = [a, b] \text{ e } f \text{ unimodal,} \end{array}$$

Algoritmo 2.1: Método de Busca Dicotômica

Entrada: Intervalo inicial de incerteza $[a, b]$, constante de diferenciação $\varepsilon > 0$, tolerância $\delta > 0$ e função objetivo f .

Saída: Minimizante x^* e mínimo $f(x^*)$.

$k := 0$

enquanto $b - a > \delta$ **faça**

$x_1 := \frac{a+b}{2} - \varepsilon$

$x_2 := \frac{a+b}{2} + \varepsilon$

se $f(x_1) < f(x_2)$ **então**

$b := x_2$

senão

$a := x_1$

fim

$k := k + 1$

fim

O intervalo final de incerteza é $[a, b]$

$x^* := \frac{a+b}{2}$

Mín: $= f(x^*)$

Observação 2.2 A precisão δ pode ser definida como a amplitude do intervalo de incerteza final (após as iterações pararem). \square

Apresenta-se a seguir uma função criada em linguagem MATLAB¹ para uma implementação computacional do Algoritmo 2.1, embora na referida função se considere como critério de paragem um determinado número de iterações e não uma precisão para o intervalo de incerteza. Designaremos tal função por *dicotomica*.

```
function dicotomica(f,a,b,n,epsilon)
%Determina o mínimo de uma função unidimensional unimodal pelo
%método de busca dicotômica
%-----DADOS DE ENTRADA-----
%f=função objetivo
%a=limite inferior do intervalo inicial de incerteza
%b=limite superior do intervalo inicial de incerteza
%n= número de iterações
%epsilon= constante de diferenciação (valor positivo e pequeno)
%Sintaxe para f: f=inline('x*sin(x)','x')
%Exemplo 1: dicotomica(inline('x*sin(x)','x'),4,5,11,0.001)
%-----DADOS DE SAÍDA-----
%minimo
```

¹Esta função foi adaptada da proposta por Aamir Din em *www.mathworks.com* e disponível em 31 de Julho de 2014.

```
%minimizante
%a=limite inferior do intervalo de incerteza final
%b=limite superior do intervalo de incerteza final
%-----
disp('Iteração      a          b          x1          x2
     f(x1)      f(x2)');
disp('-----
-----');
for i = 1:n
med = (a + b)/2;
x1 = med - epsilon;
x2 = med + epsilon;
fprintf('%2d', i);
fprintf('\t\t');
fprintf('%12.4f', a);
fprintf('\t\t');
fprintf('%12.4f', b);
fprintf('\t\t');
fprintf('%12.4f', x1);
fprintf('\t\t');
fprintf('%12.4f', x2);
fprintf('\t\t');
fprintf('%12.4f', f(x1));
fprintf('\t\t');
fprintf('%12.4f', f(x2));
fprintf('\n');
disp('-----
-----')
if (f(x1) < f(x2))
b = x2;
elseif (f(x1) > f(x2))
a = x1;
elseif (f(x1) == f(x2))
a = x1;
b = x2;
end
end
a
b
minimizante = (a+b)/2;
minimizante
minimo=f(minimizante)
end
```

Quando não for conhecida uma tolerância ou precisão para o intervalo de incerteza, em vez de determinarmos o número de iterações para aplicarmos a função *dicotomica*,

podemos usar uma outra função que nos resolverá o determinado problema considerando a tolerância dada e não o número de iterações. A essa função, em linguagem MATLAB, chamaremos *dicotomica2*², a qual apresentamos a seguir.

```
function dicotomica2(f,a,b,delta,epsilon)
%Determina o mínimo de uma função unidimensional unimodal pelo
%método de busca dicotômica
%-----DADOS DE ENTRADA-----
%f=função objetivo
%a=limite inferior do intervalo inicial de incerteza
%b=limite superior do intervalo inicial de incerteza
%delta= precisão do intervalo de incerteza
%epsilon= constante de diferenciação (valor positivo e pequeno)
%Sintaxe para f: f=inline('x*sin(x)','x')
%Exemplo 1: dicotomica(inline('x*sin(x)','x'),4,5,0.1,0.001)
%-----DADOS DE SAÍDA-----
%minimo
%minimizante
%a=limite inferior do intervalo de incerteza final
%b=limite superior do intervalo de incerteza final
%-----
disp('Iteração      a          b          x1          x2
      f(x1)      f(x2)');
disp('-----
-----');
i=1;
while b-a>delta
med = (a + b)/2;
x1 = med - epsilon;
x2 = med + epsilon;
fprintf('%2d', i);
fprintf('\t\t');
fprintf('%12.4f', a);
fprintf('\t\t');
fprintf('%12.4f', b);
fprintf('\t\t');
fprintf('%12.4f', x1);
fprintf('\t\t');
fprintf('%12.4f', x2);
fprintf('\t\t');
fprintf('%12.4f', f(x1));
fprintf('\t\t');
fprintf('%12.4f', f(x2));
```

²Esta função foi adaptada da proposta por Aamir Din em www.mathworks.com e disponível em 31 de Julho de 2014.

```
fprintf('\n');
disp('-----')
-----')
if (f(x1) < f(x2))
b = x2;
elseif (f(x1) > f(x2))
a = x1;
elseif (f(x1) == f(x2))
a = x1;
b = x2;
end
i=i+1;
end
a
b
minimizante = (a+b)/2;
minimizante
minimo=f(minimizante)
end
```

Seja $[a_0, b_0]$ o intervalo de incerteza inicial. No final da iteração k , a amplitude do intervalo de incerteza é dada por

$$b_k - a_k = \frac{b_0 - a_0}{2^k} + 2\varepsilon \left(1 - \frac{1}{2^k}\right). \quad (2.1)$$

Vamos provar, por indução, a validade da Igualdade (2.1). Para $k = 1$, resulta

$$b_1 - a_1 = \frac{b_0 - a_0}{2} + 2\varepsilon \left(1 - \frac{1}{2}\right), \text{ ou seja, } b_1 - a_1 = \frac{b_0 - a_0}{2} + \varepsilon.$$

O que é verdade, atendendo que no final da iteração k , $k \in \mathbb{N}$, a amplitude do intervalo de incerteza é dada por

$$b_k - a_k = \frac{b_{k-1} - a_{k-1}}{2} + \varepsilon, \quad (2.2)$$

como decorre da descrição feita para a primeira iteração, suportada pela Figura 2.6 (pág. 36) e pela Figura 2.7 (pág. 36).

Admitamos que a Igualdade (2.1) se verifica para um certo $k \in \mathbb{N}$. Vamos mostrar que a propriedade é hereditária, isto é,

$$b_{k+1} - a_{k+1} = \frac{b_0 - a_0}{2^{k+1}} + 2\varepsilon \left(1 - \frac{1}{2^{k+1}}\right).$$

Como

$$b_{k+1} - a_{k+1} = \frac{b_k - a_k}{2} + \varepsilon,$$

então atendendo à hipótese de indução tem-se

$$b_{k+1} - a_{k+1} = \frac{\frac{b_0 - a_0}{2^k} + 2\varepsilon \left(1 - \frac{1}{2^k}\right)}{2} + \varepsilon,$$

$$b_{k+1} - a_{k+1} = \frac{b_0 - a_0}{2^{k+1}} + \varepsilon \left(2 - \frac{2}{2^k}\right),$$

$$b_{k+1} - a_{k+1} = \frac{b_0 - a_0}{2^{k+1}} + 2\varepsilon \left(1 - \frac{1}{2^{k+1}}\right).$$

Portanto, conclui-se que a Igualdade (2.1) é hereditária. Como adicionalmente se verifica para $k = 1$, é válida para todo $k \in \mathbb{N}$, de acordo com o princípio de indução matemática.

A Fórmula (2.1) pode ser usada para determinar o número k de iterações necessárias para alcançar uma precisão desejada. Dado que em cada iteração são necessárias duas experiências³, a fórmula pode ser usada para determinar o número de experiências.

Para concluir a abordagem do método de busca dicotômica, vamos resolver alguns problemas de minimização com recurso a este método.

Problema 2.1 Considere a função f definida, em \mathbb{R} , por $f(x) = x^4 - 14x^3 + 60x^2 - 70x$.

Sabe-se que:

- a função f é unimodal, com mínimo, no intervalo $[0, 2]$;
- x^* é o minimizante de f em $[0, 2]$.

Determine uma aproximação para o mínimo de f , localizando x^* num intervalo de amplitude não superior a $\delta = 0.3$, usando o método de busca dicotômica. Utilize para constante de diferenciação $\varepsilon = 0.1$.

Resolução:

1.^a iteração

Como o intervalo de incerteza inicial é $[0, 2]$, $a = 0$ e $b = 2$. O intervalo inicial não satisfaz o critério de paragem, porque $b - a = 2 - 0 = 2 > 0.3$.

³Este termo será usado várias vezes, ao longo deste trabalho, para fazer referência ao cálculo do valor da função objetivo num ponto. O termo é usado nas aplicações pois, por vezes, é necessário fazer ensaios experimentais para calcular esse valor.

$$x_1 = \frac{a+b}{2} - \varepsilon = \frac{0+2}{2} - 0.1 = 0.9,$$

$$x_2 = \frac{a+b}{2} + \varepsilon = \frac{0+2}{2} + 0.1 = 1.1,$$

$$f(x_1) = f(0.9) = 0.9^4 - 14 \times 0.9^3 + 60 \times 0.9^2 - 70 \times 0.9 = -23.9499,$$

$$f(x_2) = f(1.1) = 1.1^4 - 14 \times 1.1^3 + 60 \times 1.1^2 - 70 \times 1.1 = -21.5699.$$

Atendendo que $f(0.9) < f(1.1)$, então elimina-se o intervalo $]1.1, 2]$. O novo intervalo de incerteza é $[0, 1.1]$.

Passamos à 2.^a iteração, porque a amplitude do intervalo de incerteza é $1.1 - 0 = 1.1$ e não satisfaz o critério de paragem.

2.^a iteração

Como o intervalo de incerteza obtido no final da 1.^a iteração é $[0, 1.1]$, $a = 0$ e $b = 1.1$.

$$x_1 = \frac{a+b}{2} - \varepsilon = \frac{0+1.1}{2} - 0.1 = 0.45,$$

$$x_2 = \frac{a+b}{2} + \varepsilon = \frac{0+1.1}{2} + 0.1 = 0.65,$$

$$f(x_1) = f(0.45) = 0.45^4 - 14 \times 0.45^3 + 60 \times 0.45^2 - 70 \times 0.45 \approx -20.5847,$$

$$f(x_2) = f(0.65) = 0.65^4 - 14 \times 0.65^3 + 60 \times 0.65^2 - 70 \times 0.65 \approx -23.8162.$$

Atendendo que $f(0.45) > f(0.65)$, então elimina-se o intervalo $[0, 0.45[$, sendo que o novo intervalo de incerteza é o intervalo $[0.45, 1.1]$. Passamos à 3.^a iteração, porque a amplitude do intervalo de incerteza é $1.1 - 0.45 = 0.65$ e não satisfaz o critério de paragem.

3.^a iteração

Nesta iteração, vamos considerar o intervalo de incerteza $[0.45, 1.1]$. Logo, $a = 0.45$ e $b = 1.1$.

$$x_1 = \frac{a+b}{2} - \varepsilon = \frac{0.45+1.1}{2} - 0.1 = 0.675,$$

$$x_2 = \frac{a+b}{2} + \varepsilon = \frac{0.45+1.1}{2} + 0.1 = 0.875,$$

$$f(x_1) = f(0.675) = 0.675^4 - 14 \times 0.675^3 + 60 \times 0.675^2 - 70 \times 0.675 \approx -24.0106,$$

$$f(x_2) = f(0.875) = 0.875^4 - 14 \times 0.875^3 + 60 \times 0.875^2 - 70 \times 0.875 \approx -24.1052.$$

Atendendo que $f(0.675) > f(0.875)$, então elimina-se o intervalo $[0.45, 0.675[$. Portanto, o novo intervalo de incerteza é o intervalo $[0.675, 1.1]$. Uma vez que a amplitude do intervalo de incerteza é $1.1 - 0.675 = 0.425$, não satisfaz o critério de paragem. Pelo que passamos a 4.^a iteração.

4.^a iteração

Nesta iteração, vamos considerar o intervalo de incerteza $[0.675, 1.1]$. Logo, $a = 0.675$ e $b = 1.1$.

$$x_1 = \frac{a+b}{2} - \varepsilon = \frac{0.675 + 1.1}{2} - 0.1 = 0.7875,$$

$$x_2 = \frac{a+b}{2} + \varepsilon = \frac{0.675 + 1.1}{2} + 0.1 = 0.9875,$$

$$f(x_1) = f(0.7875) = 0.7875^4 - 14 \times 0.7875^3 + 60 \times 0.7875^2 - 70 \times 0.7875 \approx -24.3683,$$

$$f(x_2) = f(0.9875) = 0.9875^4 - 14 \times 0.9875^3 + 60 \times 0.9875^2 - 70 \times 0.9875 \approx -23.1462.$$

Dado que $f(0.7875) < f(0.9875)$, eliminamos o intervalo $]0.9875, 1.1]$. Portanto, o novo intervalo de incerteza é o intervalo $[0.675, 0.9875]$. Uma vez que a amplitude do intervalo de incerteza é $0.9875 - 0.675 = 0.3125$, a qual não satisfaz o critério de paragem. Temos que passar para a 5.^a iteração.

5.^a iteração

O intervalo contraído obtido no final da iteração anterior é o intervalo $[0.675, 0.9875]$. Por conseguinte, nesta iteração esse será o intervalo de incerteza, pelo que $a = 0.675$ e $b = 0.9875$.

$$x_1 = \frac{a+b}{2} - \varepsilon = \frac{0.675 + 0.9875}{2} - 0.1 = 0.73125,$$

$$x_2 = \frac{a+b}{2} + \varepsilon = \frac{0.675 + 0.9875}{2} + 0.1 = 0.93125,$$

$$f(x_1) = f(0.73125) = 0.73125^4 - 14 \times 0.73125^3 + 60 \times 0.73125^2 - 70 \times 0.73125 \approx -24.2922,$$

$$f(x_2) = f(0.93125) = 0.93125^4 - 14 \times 0.93125^3 + 60 \times 0.93125^2 - 70 \times 0.93125 \approx -23.7083.$$

Dado que $f(0.73125) < f(0.93125)$, eliminamos o intervalo $]0.93125, 0.9875]$. Deste modo, o novo intervalo de incerteza é o intervalo $[0.675, 0.93125]$. Este intervalo satisfaz o critério de paragem, porque a sua amplitude é $0.93125 - 0.675 = 0.25625 < 0.3$.

Então, $x^* \approx \frac{0.675 + 0.93125}{2} = 0.803125$ e a aproximação obtida para o mínimo da função f no intervalo $[0, 2]$ é $f(0.803125) = 0.803125^4 - 14 \times 0.803125^3 + 60 \times 0.803125^2 - 70 \times 0.803125 \approx -24.3545$.

Observação 2.3 O valor exato do minimizante é 0.780884053088..., obtendo as raízes de $f'(x)$ no MATLAB com recurso ao seguinte comando: `roots([4 -42 120 -70])`. O mínimo exato é -24.369601567355... \square

Problema 2.2 Considere a função unimodal g definida, em $\left[\frac{1}{10}, 3\right]$, por

$$g(x) = 0.65 - \frac{0.75}{1+x^2} - 0.65x \arctan \frac{1}{x}.$$

Determine uma aproximação do valor x^* que minimiza g no intervalo $\left[\frac{1}{10}, 3\right]$, realizando dez iterações do método de busca dicotómica. Use para constante de diferenciação $\varepsilon = 0.00005$.

Resolução:

Para resolver este problema, vamos recorrer à função dicotómica. Para o efeito, no editor do MATLAB, basta escrever o seguinte:

```
dicotomica(inline('0.65-0.75/(1+x^2)-0.65*x*atan(1/x)', 'x'), 0.1, 3, 10, 0.00005)
```

Abaixo apresenta-se o resultado da aplicação da função dicotómica.

| Iteração | a | b | x1 | x2 | f(x1) | f(x2) |
|----------|--------|--------|--------|--------|---------|---------|
| 1 | 0.1000 | 3.0000 | 1.5499 | 1.5501 | -0.1477 | -0.1477 |
| 2 | 0.1000 | 1.5501 | 0.8250 | 0.8251 | -0.2687 | -0.2687 |
| 3 | 0.1000 | 0.8251 | 0.4625 | 0.4626 | -0.3098 | -0.3098 |
| 4 | 0.4625 | 0.8251 | 0.6437 | 0.6438 | -0.2982 | -0.2982 |
| 5 | 0.4625 | 0.6438 | 0.5531 | 0.5532 | -0.3074 | -0.3074 |
| 6 | 0.4625 | 0.5532 | 0.5078 | 0.5079 | -0.3096 | -0.3096 |

| | | | | | | |
|----|--------|--------|--------|--------|---------|---------|
| 7 | 0.4625 | 0.5079 | 0.4851 | 0.4852 | -0.3100 | -0.3100 |
| 8 | 0.4625 | 0.4852 | 0.4738 | 0.4739 | -0.3100 | -0.3100 |
| 9 | 0.4738 | 0.4852 | 0.4795 | 0.4796 | -0.3100 | -0.3100 |
| 10 | 0.4795 | 0.4852 | 0.4823 | 0.4824 | -0.3100 | -0.3100 |

$a = 0.4795$
 $b = 0.4824$
 $\text{minimizante} = 0.4809$
 $\text{minimo} = -0.3100$

Após dez iterações, envolvendo vinte experiências, o intervalo de incerteza é $[0.4795, 0.4824]$. O minimizante pode ser estimado como sendo o ponto médio 0.4809 do intervalo de incerteza.

Problema 2.3 Considere a função h definida, em $[4, 5]$, por $h(x) = x \sin x$.

Sabendo que h é unimodal com mínimo, determine o valor x^* que minimiza h , com uma precisão inferior a 0.03 , usando para constante de diferenciação $\varepsilon = 0.001$.

Resolução:

Para resolver este problema, vamos usar a função `dicotomica2`. No editor do MATLAB basta escrever

```
dicotomica2(inline('x*sin(x)','x'),4,5,0.03,0.001).
```

De onde resulta o seguinte:

| Iteração | a | b | x1 | x2 | f(x1) | f(x2) |
|----------|--------|--------|--------|--------|---------|---------|
| 1 | 4.0000 | 5.0000 | 4.4990 | 4.5010 | -4.3970 | -4.4008 |
| 2 | 4.4990 | 5.0000 | 4.7485 | 4.7505 | -4.7454 | -4.7471 |
| 3 | 4.7485 | 5.0000 | 4.8732 | 4.8753 | -4.8103 | -4.8107 |
| 4 | 4.8732 | 5.0000 | 4.9356 | 4.9376 | -4.8132 | -4.8129 |
| 5 | 4.8732 | 4.9376 | 4.9044 | 4.9064 | -4.8143 | -4.8144 |

| | | | | | | |
|---|--------|--------|--------|--------|---------|---------|
| 6 | 4.9044 | 4.9376 | 4.9200 | 4.9220 | -4.8143 | -4.8143 |
|---|--------|--------|--------|--------|---------|---------|

```

a = 4.9044
b = 4.9220
minimizante = 4.9132
minimo = -4.8145

```

Como se pode verificar pelo resultado da aplicação da função `dicotomica2`, as duas últimas observações são $x_1 = 4.9200$ e $x_2 = 4.9220$, sendo que $f(x_1) < f(x_2)$. Ao fim de seis iterações o intervalo de incerteza é $[4.9044, 4.9220]$ e o minimizante é estimado usando ponto médio 4.9132 do intervalo de incerteza. O mínimo é, então, $f(4.9132) \approx -4.8145$.

2.3 Método de Busca por Bissecção

Seja f uma função unimodal no intervalo $[a, b]$, com mínimo em x^* .

Sendo A a amplitude do intervalo de incerteza $[a, b]$, para aplicar o método de busca por bissecção, na primeira iteração, divide-se o intervalo em quatro partes de comprimento $\frac{A}{4}$, como sugere a Figura 2.8. Nessa figura, representamos o ponto médio do intervalo de incerteza por x_m e os demais pontos, que dividem o intervalo, por x_1 e x_2 , sendo $x_1 < x_2$. Seguidamente comparam-se os valores da função objetivo f em x_1 , x_m e x_2 . Dada a unimodalidade de f em $[a, b]$, apenas três casos podem ocorrer nas comparações das experiências $f(x_1)$, $f(x_m)$ e $f(x_2)$:

- (a) $f(x_1) < f(x_m) < f(x_2)$;
- (b) $f(x_2) < f(x_m) < f(x_1)$;
- (c) $f(x_1) > f(x_m)$ e $f(x_2) > f(x_m)$.

O passo seguinte corresponde à contração do intervalo de incerteza e consequente definição de um novo, eliminando-se parte do primeiro, onde garantidamente x^* não está contido.

No caso (a), como se ilustra na Figura 2.9, o novo intervalo de incerteza é $[a, x_m]$, sendo x_1 o seu ponto médio.

Se as experiências $f(x_1)$, $f(x_m)$ e $f(x_2)$ verificarem a relação (b), o novo intervalo de incerteza será $[x_m, b]$, cujo ponto médio é x_2 .

Se se verificar o caso (c), o intervalo de incerteza contraído será $[x_1, x_2]$, de ponto médio x_m .

Conhecido o novo intervalo de incerteza, em cada uma das iterações depois da primeira, procede-se à divisão deste em quatro partes iguais e repete-se tudo o que foi feito na primeira iteração. Porém, uma vez que em qualquer um dos casos atrás analisados, o ponto médio do intervalo contraído corresponde a uma experiência já calculada, passam a ser necessárias apenas duas novas experiências — correspondentes a um quarto e a três quartos do intervalo. Portanto, o método requer três experiências na primeira iteração e apenas duas novas nas iterações seguintes, sendo que reutiliza a correspondente ao ponto médio, obtida na iteração anterior.

Como foi referido atrás, o intervalo de incerteza obtido no final de cada iteração só pode ser ou $[a, x_m]$, ou $[x_m, b]$ ou $[x_1, x_2]$. Sendo a amplitude de cada um deles igual a $x_m - a = b - x_m = x_2 - x_1 = \frac{b-a}{2}$. Ou seja, em cada iteração o intervalo de incerteza reduz-se à metade.

Portanto, considerando que no final da iteração k , com $k \in \mathbb{N}$, o intervalo de incerteza é $[a_k, b_k]$ e que $[a_0, b_0]$ é o intervalo inicial,

$$b_k - a_k = \frac{1}{2}(b_{k-1} - a_{k-1}). \quad (2.3)$$

O que mostra que as amplitudes dos intervalos de incerteza estão em progressão geométrica de razão $\frac{1}{2}$ e primeiro termo $b_1 - a_1$. Logo, para todo $k \in \mathbb{N}$, tem-se

$$b_k - a_k = (b_1 - a_1) \left(\frac{1}{2}\right)^{k-1}. \quad (2.4)$$

Da Igualdade (2.3) resulta que $b_1 - a_1 = \frac{1}{2}(b_0 - a_0)$. Substituindo na Igualdade (2.4), $b_1 - a_1$ por $\frac{1}{2}(b_0 - a_0)$, obtém-se

$$b_k - a_k = (b_0 - a_0) \left(\frac{1}{2}\right)^k. \quad (2.5)$$

A Igualdade (2.5) dá-nos, então, a amplitude do intervalo de incerteza no final da iteração k , em função da amplitude do intervalo inicial. Verifica-se que

$$\lim_{k \rightarrow +\infty} (b_k - a_k) = \lim_{k \rightarrow +\infty} \left[(b_0 - a_0) \left(\frac{1}{2}\right)^k \right] = (b_0 - a_0) \times 0 = 0.$$

A tabela 2.1 mostra que no final da iteração k , são acumuladas $2k + 1$ experiências. Deste modo, sendo n o número total de experiências realizadas até ao final da iteração k , verifica-se que $n = 2k + 1$. Esta relação permite-nos calcular o número de experiências realizadas ao fim de um determinado número de iterações e vice-versa.

| | | | | | | |
|--|---|---|---|---|-----|----------|
| N. ^o de iterações | 1 | 2 | 3 | 4 | ... | k |
| N. ^o de experiências acumuladas | 3 | 5 | 7 | 9 | ... | $2k + 1$ |

Tabela 2.1: Número total de experiências em função do número de iterações.

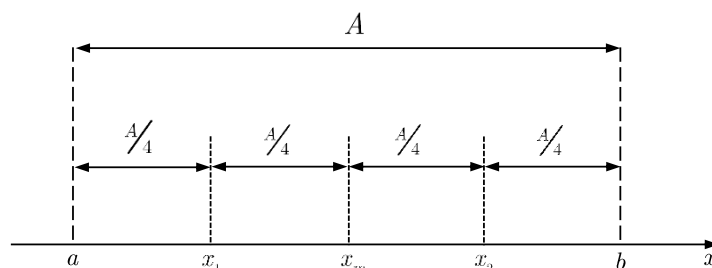
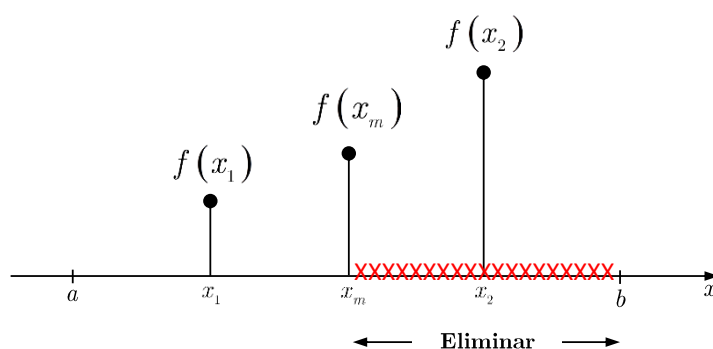


Figura 2.8: Divisão do intervalo de incerteza em quatro partes iguais.

Figura 2.9: Possibilidades no método de busca por bissecção. Caso $f(x_1) < f(x_m) < f(x_2)$.

O Algoritmo 2.2 resume o método aqui analisado e servirá de base à implementação computacional, em linguagem **MATLAB**, de uma função que usaremos para resolver alguns problemas. Designaremos tal função por *bisseccao* e apresentámo-la a seguir.

```
function bisseccao(f,a,b,delta)
%Determina o mínimo de uma função unidimensional unimodal pelo
%método de busca por bissecção
%-----DADOS DE ENTRADA-----
%f=função objetivo
%a=limite inferior do intervalo inicial de incerteza
%b=limite superior do intervalo inicial de incerteza
%delta= precisão do intervalo de incerteza
%Syntaxe para f: f=inline('x*sin(x)','x')
%Exemplo 1: bisseccao(inline('x*sin(x)','x'),4,5,0.001)
```

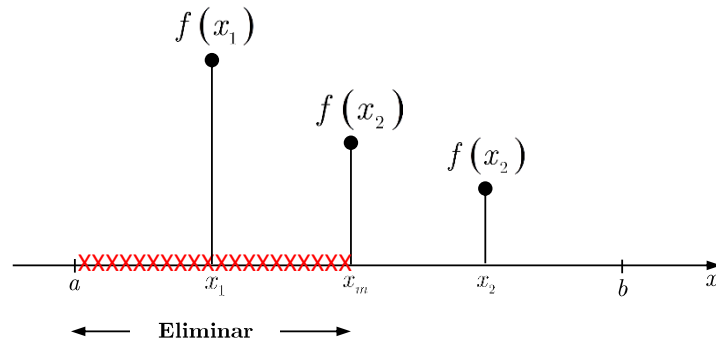



Figura 2.10: Possibilidades no método de busca por bissecção. Caso $f(x_2) < f(x_m) < f(x_1)$.

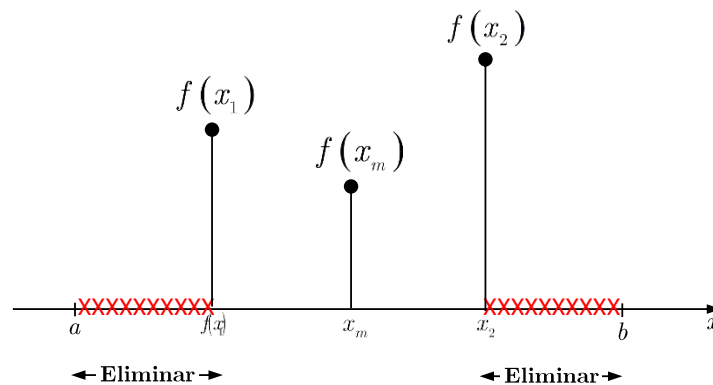


Figura 2.11: Possibilidades no método de busca por bissecção. Caso $f(x_1) > f(x_m)$ e $f(x_2) > f(x_m)$.

```
%-----DADOS DE SAÍDA-----
%minimo
%minimizante
%a=limite inferior do intervalo de incerteza final
%b=limite superior do intervalo de incerteza final
%-----
disp('k      a      x1      xm      x2      b
f(x1)      f(xm)      f(x2)');
disp('-----
-----');
i=1;
while b-a>delta
```

Algoritmo 2.2: Método de Busca por Bissecção.

Entrada: Intervalo inicial de incerteza $[a, b]$, tolerância $\delta > 0$ e função objetivo f .

Saída: Intervalo final de incerteza, minimizante x^* e mínimo $f(x^*)$.

$k := 0$

enquanto $b - a > \delta$ **faça**

$x_1 := a + \frac{b - a}{4}$

$x_m := a + 2 \times \frac{b - a}{4}$

$x_2 := a + 3 \times \frac{b - a}{4}$

se $f(x_1) < f(x_m)$ e $f(x_m) < f(x_2)$ **então**

$b := x_m$

senão

se $f(x_2) < f(x_m)$ e $f(x_m) < f(x_1)$ **então**

$a := x_m$

senão se $f(x_1) > f(x_m)$ e $f(x_2) > f(x_m)$ **então**

$a := x_1$

$b := x_2$

fim

fim

$k := k + 1$

fim

O intervalo de incerteza final é $[a, b]$, $x^* := \frac{a + b}{2}$ e Mín: $= f(x^*)$

```
x1 =a +(1/4)*(b-a);
xm=a+(2/4)*(b-a);
x2 = a+(3/4)*(b-a);
fprintf('%2d', i);
fprintf('\t');
fprintf('%7.4f', a);
fprintf('\t');
fprintf('%7.4f', x1);
fprintf('\t');
fprintf('%8.4f', xm);
fprintf('\t');
fprintf('%8.4f', x2);
fprintf('\t');
fprintf('%8.4f', b);
fprintf('\t');
fprintf('%8.4f', f(x1));
fprintf('\t');
fprintf('%8.4f', f(xm));
```

```

fprintf('t');
fprintf('%8.4f', f(x2));
fprintf('\n');
disp('-----')
-----')
if (f(x1) < f(xm) & f(xm) < f(x2))
b = xm;
elseif (f(x2) < f(xm) & f(xm) < f(x1))
a = xm;
elseif (f(x1) > f(xm) & f(x2) > f(xm))
a = x1;
b = x2;
end
i=i+1;
end
fprintf('0 limite inferior do intervalo de incerteza final é a= %.4f\n',a);
fprintf('0 limite superior do intervalo de incerteza final é b= %.4f\n',b);
fprintf('Uma aproximação para o minimizante é x^*=%.4f\n',(a+b)/2);
fprintf('Uma aproximação para o mínimo é Min= %.4f\n',f((a+b)/2));
end

```

Vamos seguidamente resolver alguns problemas de minimização, usando, para o efeito, o método que acabamos de apresentar. Nos dois últimos vamos apenas usar a função *bisseccao*, enquanto que o primeiro será resolvido de forma manual com recurso a uma calculadora.

Problema 2.4 *Seja f a função definida, em \mathbb{R} , por $f(x) = e^{3x} + 5e^{-2x}$.*

Considere que:

- *a função f é unimodal, com mínimo, no intervalo $[0, 1]$;*
- *x^* é o minimizante de f em $[0, 1]$.*

Determine uma aproximação para o mínimo de f no intervalo $[0, 1]$, realizando três iterações do método de busca por bissecção.

Resolução:

1.^a iteração

Como o intervalo de incerteza inicial é $[0, 1]$, $a = 0$ e $b = 1$.

$$x_1 = a + \frac{1}{4}(b - a) = 0 + \frac{1}{4} \times (1 - 0) = 0.25,$$

$$x_m = a + \frac{2}{4}(b - a) = 0 + \frac{2}{4} \times (1 - 0) = 0.5,$$

$$x_2 = a + \frac{3}{4}(b - a) = 0 + \frac{3}{4} \times (1 - 0) = 0.75,$$

$$f(x_1) = f(0.25) = e^{3 \times 0.25} + 5e^{-2 \times 0.25} \approx 5.1497,$$

$$f(x_m) = f(0.5) = e^{3 \times 0.5} + 5e^{-2 \times 0.5} \approx 6.3211,$$

$$f(x_2) = f(0.75) = e^{3 \times 0.75} + 5e^{-2 \times 0.75} \approx 10.6034.$$

Atendendo que $f(x_1) < f(x_m) < f(x_2)$, então elimina-se o intervalo $]x_m, b] =]0.5, 1]$.

O novo intervalo de incerteza é, então, $[a, x_m] = [0, 0.5]$.

2.^a iteração

Como o intervalo de incerteza obtido no final da 1.^a iteração é $[0, 0.5]$, $a = 0$ e $b = 0.5$.

$$x_1 = a + \frac{1}{4}(b - a) = 0 + \frac{1}{4} \times (0.5 - 0) = 0.125,$$

$$x_m = a + \frac{2}{4}(b - a) = 0 + \frac{2}{4} \times (0.5 - 0) = 0.25,$$

$$x_2 = a + \frac{3}{4}(b - a) = 0 + \frac{3}{4} \times (0.5 - 0) = 0.375,$$

$$f(x_1) = f(0.125) = e^{3 \times 0.125} + 5e^{-2 \times 0.125} \approx 5.349,$$

$$f(x_m) = f(0.25) = e^{3 \times 0.25} + 5e^{-2 \times 0.25} \approx 5.1497,$$

$$f(x_2) = f(0.375) = e^{3 \times 0.375} + 5e^{-2 \times 0.375} \approx 5.442.$$

Uma vez que $f(x_1) > f(x_m)$ e $f(x_2) > f(x_m)$, eliminaremos os intervalos $[a, x_1[= [0, 0.125[$ e $]x_2, b] =]0.375, 0.5]$. O novo intervalo de incerteza é, então, $[x_1, x_2] = [0.125, 0.375]$.

3.^a iteração

Nesta última iteração, vamos considerar o intervalo de incerteza $[0.125, 0.375]$. Logo, $a = 0.125$ e $b = 0.375$.

$$x_1 = a + \frac{1}{4}(b - a) = 0.125 + \frac{1}{4} \times (0.375 - 0.125) = 0.1875,$$

$$x_m = a + \frac{2}{4}(b - a) = 0.125 + \frac{2}{4} \times (0.375 - 0.125) = 0.25,$$

$$x_2 = a + \frac{3}{4}(b - a) = 0.125 + \frac{3}{4} \times (0.375 - 0.125) = 0.3125,$$

$$f(x_1) = f(0.1875) = e^{3 \times 0.1875} + 5e^{-2 \times 0.1875} \approx 5.1915,$$

$$f(x_m) = f(0.25) = e^{3 \times 0.25} + 5e^{-2 \times 0.25} \approx 5.1497,$$

$$f(x_2) = f(0.3125) = e^{3 \times 0.3125} + 5e^{-2 \times 0.3125} \approx 5.2299.$$

Uma vez que $f(x_1) > f(x_m)$ e $f(x_2) > f(x_m)$, são eliminados os intervalos $[a, x_1[= [0.125, 0.1875[$ e $]x_2, b] =]0.3125, 0.375]$. Portanto, o intervalo final de incerteza é $[x_1, x_2] = [0.1875, 0.3125]$.

Considerando $a = x_1 = 0.1875$ e $b = x_2 = 0.3125$, uma aproximação para o mínimo é $f\left(\frac{a+b}{2}\right) = f\left(\frac{0.1875+0.3125}{2}\right) = e^{3 \times 0.25} + 5e^{-2 \times 0.25} \approx 5.1497$.

Problema 2.5 Considere a função g definida, em $]2, 10[$, por

$$g(x) = \ln^2(x-2) + \ln^2(10-x) - x^{0.2}.$$

Sabendo que a função g é unimodal no intervalo $[6, 9.9]$, determine uma aproximação do valor x^* que minimiza g nesse intervalo, obtendo um intervalo de incerteza de amplitude inferior a 10^{-3} .

Resolução:

Para resolver este problema, vamos recorrer à função bisseccao. Para o efeito, no editor do MATLAB, basta escrever e executar a seguinte instrução:

```
bisseccao(inline('(log(x-2))^2+(log(10-x))^2-x^(0.2)', 'x'), 6, 9.9, 10^(-3))
```

Abaixo apresenta-se o resultado da aplicação da função bisseccao.

| k | a | x1 | xm | x2 | b | f(x1) | f(xm) | f(x2) |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 1 | 6.0000 | 6.9750 | 7.9500 | 8.9250 | 9.9000 | 2.3247 | 2.1820 | 2.2007 |
| 2 | 6.9750 | 7.4625 | 7.9500 | 8.4375 | 8.9250 | 2.2552 | 2.1820 | 2.1348 |
| 3 | 7.9500 | 8.1937 | 8.4375 | 8.6813 | 8.9250 | 2.1519 | 2.1348 | 2.1432 |

| | | | | | | | | |
|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 4 | 8.1937 | 8.3156 | 8.4375 | 8.5594 | 8.6813 | 2.1411 | 2.1348 | 2.1347 |
| 5 | 8.4375 | 8.4984 | 8.5594 | 8.6203 | 8.6813 | 2.1338 | 2.1347 | 2.1377 |
| 6 | 8.4375 | 8.4680 | 8.4984 | 8.5289 | 8.5594 | 2.1341 | 2.1338 | 2.1340 |
| 7 | 8.4680 | 8.4832 | 8.4984 | 8.5137 | 8.5289 | 2.1339 | 2.1338 | 2.1339 |
| 8 | 8.4832 | 8.4908 | 8.4984 | 8.5061 | 8.5137 | 2.1339 | 2.1338 | 2.1338 |
| 9 | 8.4908 | 8.4946 | 8.4984 | 8.5022 | 8.5061 | 2.1339 | 2.1338 | 2.1338 |
| 10 | 8.4984 | 8.5003 | 8.5022 | 8.5042 | 8.5061 | 2.1338 | 2.1338 | 2.1338 |
| 11 | 8.5003 | 8.5013 | 8.5022 | 8.5032 | 8.5042 | 2.1338 | 2.1338 | 2.1338 |
| 12 | 8.5003 | 8.5008 | 8.5013 | 8.5018 | 8.5022 | 2.1338 | 2.1338 | 2.1338 |
| 0 limite inferior do intervalo de incerteza final é a= 8.5013 | | | | | | | | |
| 0 limite superior do intervalo de incerteza final é b= 8.5022 | | | | | | | | |
| Uma aproximação para o minimizante é x*= 8.5018 | | | | | | | | |
| Uma aproximação para o mínimo é Min= 2.1338 | | | | | | | | |

Foram necessárias doze iterações, envolvendo vinte e cinco experiências.

Problema 2.6 Considere a função h definida, em \mathbb{R} , por $h(x) = -3x \sin(0.75x) + e^{-2x}$.

No intervalo $[0, 2\pi]$ a função h é unimodal. Determine uma aproximação para o mínimo de h nesse intervalo, com precisão inferior a 10^{-2} .

Resolução:

A resolução deste problema será feito com recurso à função bisseccao. No editor do MATLAB, basta escrever e executar a seguinte instrução:

```
bisseccao(inline('-3*x*sin(0.75*x+exp(-2*x))','x'),0,2*pi,10^(-2)).
```

Em consequência, obtém-se o seguinte resultado:

| k | a | x1 | xm | x2 | b | f(x1) | f(xm) | f(x2) |
|---|--------|--------|--------|--------|--------|---------|---------|---------|
| 1 | 0.0000 | 1.5708 | 3.1416 | 4.7124 | 6.2832 | -4.4275 | -6.6519 | 5.4111 |
| 2 | 1.5708 | 2.3562 | 3.1416 | 3.9270 | 4.7124 | -6.9201 | -6.6519 | -2.2939 |

| | | | | | | | | |
|----|--------|--------|--------|--------|--------|---------|---------|---------|
| 3 | 1.5708 | 1.9635 | 2.3562 | 2.7489 | 3.1416 | -5.8724 | -6.9201 | -7.2569 |
| 4 | 2.3562 | 2.5525 | 2.7489 | 2.9452 | 3.1416 | -7.1942 | -7.2569 | -7.0823 |
| 5 | 2.5525 | 2.6507 | 2.7489 | 2.8471 | 2.9452 | -7.2538 | -7.2569 | -7.2006 |
| 6 | 2.6507 | 2.6998 | 2.7489 | 2.7980 | 2.8471 | -7.2626 | -7.2569 | -7.2364 |
| 7 | 2.6507 | 2.6753 | 2.6998 | 2.7243 | 2.7489 | -7.2600 | -7.2626 | -7.2616 |
| 8 | 2.6753 | 2.6875 | 2.6998 | 2.7121 | 2.7243 | -7.2618 | -7.2626 | -7.2626 |
| 9 | 2.6875 | 2.6937 | 2.6998 | 2.7059 | 2.7121 | -7.2623 | -7.2626 | -7.2627 |
| 10 | 2.6998 | 2.7029 | 2.7059 | 2.7090 | 2.7121 | -7.2627 | -7.2627 | -7.2627 |

O limite inferior do intervalo de incerteza final é a= 2.7029
 O limite superior do intervalo de incerteza final é b= 2.7090
 Uma aproximação para o minimizante é $x^*= 2.7059$
 Uma aproximação para o mínimo é Min= -7.2627

Para alcançar a precisão pedida, foram necessárias dez iterações, envolvendo vinte e uma experiências.

2.4 Método de Fibonacci

Assumamos que f é uma função unimodal no intervalo $[a_0, b_0]$ e que x^* é o seu único minimizante em $[a_0, b_0]$. No final da iteração k , $k \in \mathbb{N}$, designaremos o intervalo de incerteza por $[a_k, b_k]$.

Na Secção 2.1, ficou esclarecido que se torna necessário, em cada iteração, ter pelo menos duas experiências da função objetivo, em pontos interiores do intervalo de incerteza, para se conseguir reduzir o intervalo. No método que aqui analisaremos, teremos que determinar dois pontos intermédios⁴ x_1 e x_2 ($x_1 < x_2$), no intervalo de incerteza, para avaliar a função f . A determinação dos pontos intermédios será feita, escolhendo, na iteração k , um valor λ_k , $0 \leq \lambda_k \leq \frac{1}{2}$, de modo que a partir da segunda iteração seja necessária apenas uma nova experiência, sendo que a outra será reutilizada da iteração anterior. Note-se que, deste modo, melhora-se a eficiência computacional alcançada pelos métodos de busca dicotómica

⁴Embora os pontos intermédios variem de iteração para iteração, em cada iteração vamos designá-los quase sempre por x_1 e x_2 . Excecionalmente, será usada a notação $x_{1,k}$ e $x_{2,k}$, onde k representa o número da iteração.

e de busca por bissecção, no que diz respeito ao número de novas experiências. Por outro lado, como sugere a Figura 2.12, a redução do intervalo deve ser simétrica, isto é,

$$x_1 - a_k = \lambda_k(b_k - a_k) \text{ e } b_k - x_2 = \lambda_k(b_k - a_k). \quad (2.6)$$

Portanto,

$$x_1 = a_k + \lambda_k(b_k - a_k) \text{ e } x_2 = b_k - \lambda_k(b_k - a_k). \quad (2.7)$$

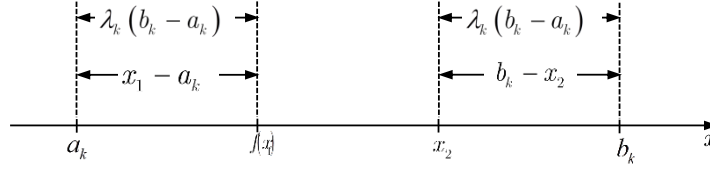


Figura 2.12: Escolha dos pontos x_1 e x_2 para avaliar f .

Com auxílio da Figura 2.13 e da Figura 2.14, vamos relacionar os valores λ_k e λ_{k+1} . Não será demais recordar que o nosso propósito é, conhecido λ_k , determinar λ_{k+1} de modo que apenas uma nova experiência seja necessária em cada iteração a partir da segunda.

Vamos começar por deduzir a relação entre as amplitude dos intervalos de incerteza de iterações consecutivas. Das figuras 2.13 e 2.14, resulta

$$b_{k+1} - a_{k+1} = \lambda_k(b_k - a_k) + (b_k - a_k) - 2\lambda_k(b_k - a_k).$$

E, portanto,

$$b_{k+1} - a_{k+1} = (\lambda_k + 1 - 2\lambda_k)(b_k - a_k),$$

$$b_{k+1} - a_{k+1} = (1 - \lambda_k)(b_k - a_k). \quad (2.8)$$

Estamos, agora, em condições de deduzir a relação entre λ_k e λ_{k+1} . Atendendo às figuras atrás referidas, conclui-se que

$$\lambda_{k+1}(b_{k+1} - a_{k+1}) = (b_k - a_k) - 2\lambda_k(b_k - a_k).$$

Daqui, e tendo em conta a Igualdade (2.8), obtém-se

$$\begin{aligned}
 \lambda_{k+1}(1 - \lambda_k)(b_k - a_k) &= (1 - 2\lambda_k)(b_k - a_k), \\
 \lambda_{k+1}(1 - \lambda_k) &= (1 - 2\lambda_k), \\
 \lambda_{k+1} &= \frac{1 - 2\lambda_k}{1 - \lambda_k}, \\
 \lambda_{k+1} &= \frac{1 - \lambda_k}{1 - \lambda_k} + \frac{-\lambda_k}{1 - \lambda_k}, \\
 \lambda_{k+1} &= 1 - \frac{\lambda_k}{1 - \lambda_k}.
 \end{aligned} \tag{2.9}$$

Aplicando a Igualdade (2.8), podemos concluir que ao fim de n iterações a amplitude do intervalo de incerteza é

$$b_n - a_n = (1 - \lambda_1)(1 - \lambda_2) \cdots (1 - \lambda_n)(b_0 - a_0). \tag{2.10}$$

Deste modo, estamos interessados numa sequência de n valores $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ tais que $0 \leq \lambda_k \leq \frac{1}{2}$, para $k \in \{1, 2, \dots, n\}$, $\lambda_{k+1} = 1 - \frac{\lambda_k}{1 - \lambda_k}$, para $k \in \{1, 2, \dots, n-1\}$ e o fator de redução do intervalo de incerteza $(1 - \lambda_1)(1 - \lambda_2) \cdots (1 - \lambda_n)$ seja mínimo. De acordo com o Teorema A.1 (pág. 151),

$$\lambda_k = 1 - \frac{F_{n-k+1}}{F_{n-k+2}}, \quad k \in \{1, 2, \dots, n\}. \tag{2.11}$$

Tendo em consideração a Igualdade (2.11) e a sucessão de Fibonacci (ver Definição (A.1), pág. 144), o fator de redução do intervalo de incerteza, após n iterações, é

$$(1 - \lambda_1)(1 - \lambda_2) \cdots (1 - \lambda_n) = \frac{F_n}{F_{n+1}} \times \frac{F_{n-1}}{F_n} \times \frac{F_{n-2}}{F_{n-1}} \times \cdots \times \frac{F_1}{F_2} = \frac{1}{F_{n+1}}. \tag{2.12}$$

Portanto,

$$\frac{b_n - a_n}{b_0 - a_0} = \frac{1}{F_{n+1}}. \tag{2.13}$$

Pelo que dada uma precisão δ para o intervalo de incerteza, o número n de iterações necessárias é tal que

$$\frac{b_0 - a_0}{F_{n+1}} \leq \delta. \tag{2.14}$$

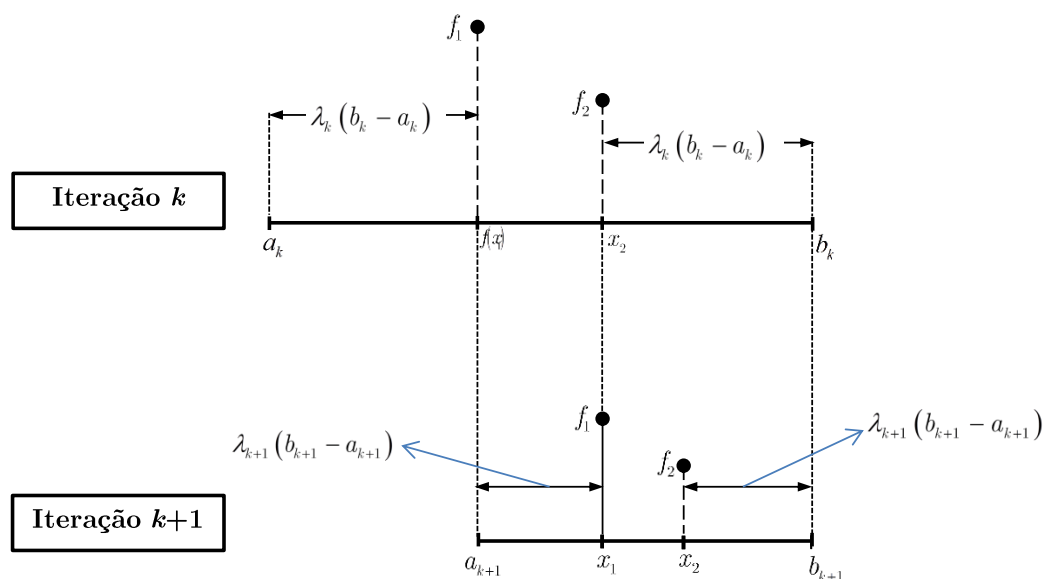


Figura 2.13: Dedução da relação entre λ_k e λ_{k+1} . Caso $f(x_1) > f(x_2)$ na iteração k .

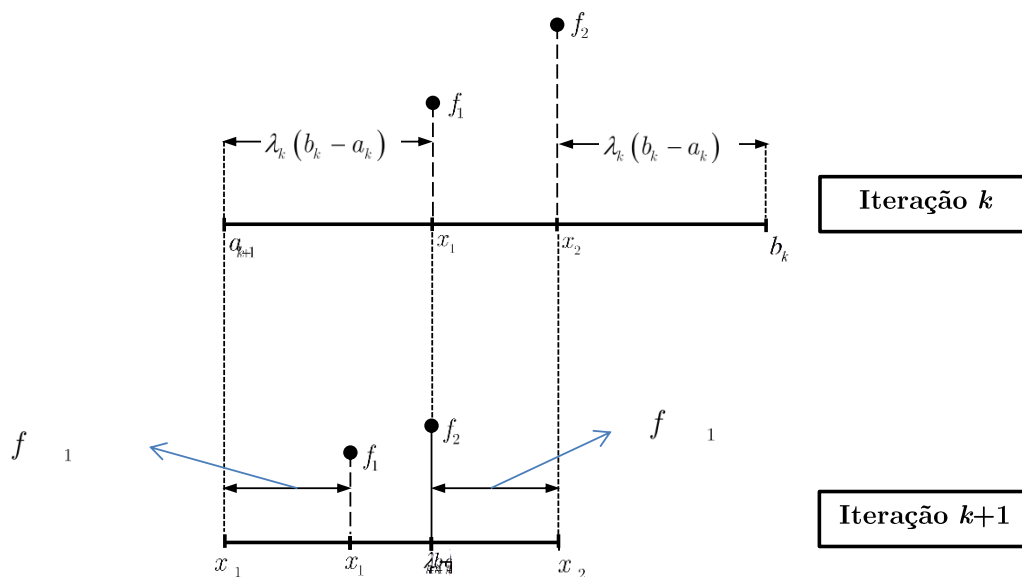


Figura 2.14: Dedução da relação entre λ_k e λ_{k+1} . Caso $f(x_1) < f(x_2)$ na iteração k .

Convém realçar que precisamos de dois pontos intermédios em cada iteração. Havendo

uma iteração anterior, um dos pontos é reutilizado dessa iteração — trata-se daquele que está no interior do intervalo de incerteza obtido no final da iteração anterior — e o outro é novo, sendo calculado usando uma das equações (2.7). Sejam $x_{1,k}$ e $x_{2,k}$ ($x_{1,k} < x_{2,k}$) os pontos intermédios obtidos na iteração k . Se $x_{1,k} \in]a_{k+1}, b_{k+1}[$, sendo $[a_{k+1}, b_{k+1}]$ o intervalo de incerteza no início da iteração $k+1$, este ponto será reutilizado nesta iteração. Nesse caso, $x_{2,k+1} = x_{1,k}$ e a nova experiência será obtida para $x_{1,k+1} = a_{k+1} + \lambda_{k+1}(b_{k+1} - a_{k+1})$. Caso ocorra $x_{2,k} \in]a_{k+1}, b_{k+1}[$, sendo $[a_{k+1}, b_{k+1}]$ o intervalo de incerteza no início da iteração $k+1$, este ponto será aproveitado nesta iteração. Nesse caso, $x_{1,k+1} = x_{2,k}$ e o novo ponto intermédio será $x_{2,k+1} = b_{k+1} - \lambda_{k+1}(b_{k+1} - a_{k+1})$.

Na última iteração, quando $k = n$, usa-se uma iteração do método da busca dicotômica, dado que, qualquer que seja o n ,

$$\lambda_n = 1 - \frac{F_1}{F_2} = \frac{1}{2}.$$

Portanto, no início da iteração n , o intervalo de incerteza resultante da iteração anterior tem como ponto médio o ponto intermédio que será reutilizado na iteração final. Sendo exatamente essa a posição do novo ponto intermédio. Basta atender que na iteração n se tem

$$x_1 = a_n + \left(1 - \frac{F_1}{F_2}\right)(b_n - a_n) = \frac{a_n + b_n}{2},$$

$$x_2 = b_n - \left(1 - \frac{F_1}{F_2}\right)(b_n - a_n) = \frac{a_n + b_n}{2}.$$

Assim sendo, os dois pontos intermédios coincidem com o ponto médio do intervalo de incerteza no início da iteração n , pelo que não é possível reduzi-lo. Para contornar este problema, nesta iteração — a última — consideramos uma constante de diferenciação positiva θ , suficientemente pequena e escolhemos o intervalo de incerteza final de acordo com o seguinte critério:

- se $f(x_1) > f(x_1 + \theta)$, o intervalo final é $[x_1, b_n]$;
- se $f(x_1) < f(x_1 + \theta)$, o intervalo final é $[a_n, x_1]$.

Esta solução para a dificuldade surgida na última iteração, consubstancia uma iteração do método de busca dicotômica e permite reduzir a amplitude do intervalo de incerteza para quase metade.

Uma desvantagem do método tem a ver com o facto de, para um número n de iterações muito elevado, a diferença, em valor absoluto, entre x_1 e x_2 poder tornar-se muito pequena e até ocorrer $x_1 > x_2$, devido a erros de arredondamento. Nesse caso, termina-se o processo iterativo e toma-se $x^* \approx x_1$ ou $x^* \approx x_2$.

Algoritmo 2.3: Método de Fibonacci

Entrada: Intervalo inicial de incerteza $[a, b]$,
constante de diferenciação $\theta > 0$,
número de iterações n ,
função objetivo f .
 $k = 1; j = 2$
 $F_0 = F_1 = 1$
 $\lambda_k := 1 - \frac{F_{n-k+1}}{F_{n-k+2}}$

repita

$F_j = F_{j-1} + F_{j-2}$
 $j := j + 1$

até $j = n + 1$;

$x_1 := a + \lambda_k (b - a)$

$x_2 := b - \lambda_k (b - a)$

enquanto $k \neq n$ e $x_1 < x_2$ **faça**

se $f(x_1) < f(x_2)$ **então**

$b := x_2$

$x_2 := x_1$

$x_1 := a + \lambda_k (b - a)$

senão

$a := x_1$

$x_1 := x_2$

$x_2 := b - \lambda_k (b - a)$

fim

$k := k + 1$

fim

se $k = n$ **então**

$x_1 := a + \lambda_k (b - a)$

$x_2 := x_1 + \theta$

se $f(x_1) > f(x_2)$ **então**

$a := x_1$

senão se $f(x_1) < f(x_2)$ **então**

$b = x_1$

fim

senão se $x_1 > x_2$ **então**

$a := x_2$

$b := x_1$

fim

O intervalo de incerteza final é $[a, b]$

$x^* := \frac{a + b}{2}$

$\text{Min} := f(x^*)$

Depois de toda esta discussão, resumimos o método de Fibonacci no Algoritmo 2.3.

Vejam os, seguidamente, alguns problemas envolvendo o método aqui discutido.

Problema 2.7 Considere que a função definida por $f(x) = \frac{\sin \frac{1}{x}}{(x - 0.2)^2 + 0.1}$ é unimodal no intervalo $[0.13, 0.4]$. Determine o mínimo de f em $[0.13, 0.4]$, realizando três iterações do método de Fibonacci.

Resolução:

1.^a **iteração** ($k = 1$)

Como o intervalo de incerteza inicial é $[0.13, 0.4]$, $a = 0.13$ e $b = 0.4$.

$$\lambda_k = 1 - \frac{F_{n-k+1}}{F_{n-k+2}}, \text{ com } k = 1 \text{ e } n = 3,$$

$$\lambda_1 = 1 - \frac{F_3}{F_4} = 1 - \frac{3}{5} = \frac{2}{5},$$

$$x_1 = a + \lambda_1(b - a) = 0.13 + \frac{2}{5} \times (0.4 - 0.13) = 0.238,$$

$$x_2 = b - \lambda_1(b - a) = 0.4 - \frac{2}{5} \times (0.4 - 0.13) = 0.292,$$

$$f(x_1) = f(0.238) = \frac{\sin \frac{1}{0.238}}{(0.238 - 0.2)^2 + 0.1} \approx -8.599804,$$

$$f(x_2) = f(0.292) = \frac{\sin \frac{1}{0.292}}{(0.292 - 0.2)^2 + 0.1} \approx -2.575047.$$

Como $f(x_1) < f(x_2)$, então elimina-se o intervalo $[x_2, b] = [0.292, 0.4]$. O novo intervalo de incerteza é, portanto, $[a, x_2] = [0.13, 0.292]$.

2.^a **iteração** ($k = 2$)

Como o intervalo de incerteza obtido no final da 1.^a iteração é $[0.13, 0.292]$, $a = 0.13$ e $b = 0.292$.

$$\lambda_k = 1 - \frac{F_{n-k+1}}{F_{n-k+2}}, \text{ com } k = 2 \text{ e } n = 3,$$

$$\lambda_2 = 1 - \frac{F_2}{F_3} = 1 - \frac{2}{3} = \frac{1}{3},$$

$$x_1 = a + \lambda_2(b - a) = 0.13 + \frac{1}{3} \times (0.31 - 0.13) = 0.184,$$

$$x_2 = 0.238 \text{ (reutilização do valor } x_1 \text{ da 1.ª iteração),}$$

$$f(x_1) = f(0.184) = \frac{\sin \frac{1}{0.184}}{(0.184 - 0.2)^2 + 0.1} \approx -7.483096,$$

$$f(x_2) = f(0.238) \approx -8.599804 \text{ (ver 1.ª iteração).}$$

Uma vez que $f(x_1) > f(x_2)$, deve ser eliminado o intervalo $[a, x_1[= [0.13, 0.184[$. Logo, o novo intervalo de incerteza é $[x_1, b] = [0.184, 0.292]$.

3.ª iteração ($k = 3$)

Nesta iteração final, vamos considerar o intervalo de incerteza $[0.184, 0.292]$. Logo, $a = 0.184$ e $b = 0.292$.

$$x_1 = 0.238 \text{ (reutilização do valor } x_2 \text{ da iteração anterior),}$$

$$\text{Consideremos } \theta = \frac{b - a}{100} = \frac{0.292 - 0.184}{100} = 0.00108$$

$$x_2 = x_1 + \theta = 0.238 + 0.00108 = 0.23908.$$

$$f(x_1) = f(0.238) = \frac{\sin \frac{1}{0.238}}{(0.238 - 0.2)^2 + 0.1} \approx -8.599804,$$

$$f(x_2) = f(0.23908) = \frac{\sin \frac{1}{0.23908}}{(0.23908 - 0.2)^2 + 0.1} \approx -8.499831,$$

Uma vez que $f(x_1) < f(x_2)$, então elimina-se o intervalo $]x_1, b] =]0.238, 0.292]$. Assim, obtém-se o intervalo de incerteza final $[a, x_1] = [0.184, 0.238]$.

Uma aproximação para o mínimo é

$$f\left(\frac{0.184 + 0.238}{2}\right) = f(0.211) = \frac{\sin \frac{1}{0.211}}{(0.211 - 0.2)^2 + 0.1} \approx -9.984288.$$

Observação 2.4 *Se no fim deste exemplo pretendermos melhorar a precisão, apertando mais o intervalo de incerteza, teremos de começar de novo redefinindo o número de iterações. Ou seja, o método de Fibonacci não permite usar um critério de paragem a posteriori.* \square

Problema 2.8 *Implemente em linguagem MATLAB o algoritmo do método de Fibonacci.*

Resolução:

Vamos adotar uma solução que após à sua aplicação devolva alguns cálculos intermédios realizados, registando-os numa tabela. Assim propomos a função fibonacci apresentada a seguir.

```
function fibonacci(f,a,b,n)
%Determina o mínimo de uma função unidimensional unimodal pelo método de
%Fibonacci
%-----DADOS DE ENTRADA-----
%f=função objetivo
%a=limite inferior do intervalo inicial de incerteza
%b=limite superior do intervalo inicial de incerteza
%n= número de iterações
%Sintaxe para f: f=inline('x*sin(x)','x')
%Exemplo: fibonacci(inline('x*sin(x)','x'),4,5,10)
%-----DADOS DE SAÍDA-----
%a=limite inferior do intervalo de incerteza final
%b=limite superior do intervalo de incerteza final
%minimizante x^*
%minimo Min
%----- INÍCIO -----
clearvars -global
F=zeros(1,n);
F(1)=1;
F(2)=2;
for j=3:(n+1)
    F(j)=F(j-1)+F(j-2);
end
k=1;
lambda = 1-(F(n-k+1))/(F(n-k+2));
x1 = a+lambda*(b-a);
x2 = b-lambda*(b-a);
fprintf('-----\n');
fprintf('k      x1      x2      f(x1)      f(x2)\n');
    Novo intervalo\n');
```

```

fprintf('-----
-----\n');
k=2;
while (k<=n & x1<x2)
    z(1)=x1;
    z(2)=x2;
    z(3)=f(x1);
    z(4)=f(x2);
    if (f(x1)<f(x2))
        b = x2;
        x2 = x1;
        lambda = 1-(F(n-k+1))/(F(n-k+2));%
        x1 = a + lambda*(b-a);
    elseif(f(x1)>f(x2))
        a = x1;
        x1 = x2;
        lambda = 1-(F(n-k+1))/(F(n-k+2));%
        x2 = b-lambda*(b-a);
    end
    z(5)=a;
    z(6)=b;
    fprintf('%2d\t %.6f\t %.6f\t %.6f\t %.6f\t',k-1, z(1), z(2), z(3), z(4));
    fprintf('%c %.6f %c %.6f %c\n', '[' ,z(5), ', ', z(6), ']' );
    k=k+1;
end
if (k-1==n)
    teta=(b-a)/4;%
    k=n;
    lambda = 1-(F(n-k+1))/(F(n-k+2));%
    x1 = a + lambda*(b-a);
    x2=x1+teta;
    if(f(x1)>f(x2))
        a=x1;
    elseif(f(x1)<f(x2))
        b=x1;
    end
elseif(x1>x2)
    a=x2;
    b=x1;
end
fprintf('%2d\t %.6f\t %.6f\t %.6f\t %.6f\t',k, x1, x2, f(x1), f(x2));
fprintf('%c %.6f %c %.6f %c\n', '[' ,a, ', ', b, ']' );
fprintf('\n');
fprintf('0 intervalo de incerteza final é ');
fprintf('%c %.6f %c %.6f %c\n', '[' ,a, ', ', b, ']' );

```



```
fprintf('Uma aproximação para o minimizante é x^*=% .6f\n',(a+b)/2);
fprintf('Uma aproximação para o mínimo é f(x^*)= %.6f\n',f((a+b)/2));
end
```

Problema 2.9 Seja g a função definida, em \mathbb{R} , por $g(x) = x^2 - \sin x$.

Sabe-se que g é unimodal no intervalo $[0, 1]$. Determine um valor aproximado para o mínimo de g , localizando o minimizante num intervalo de amplitude inferior a 0.0001. Utilize a função proposta no Problema 2.8.

Resolução:

Começemos por determinar o número n de iterações necessárias. O número n é tal que

$$b_n - a_n < 0.0001,$$

$$\frac{1}{F_{n+1}}(b_0 - a_0) < 0.0001,$$

$$F_{n+1} > \frac{1 - 0}{0.0001},$$

$$F_{n+1} > 10000.$$

O menor número natural que satisfaz a condição anterior é $n = 19$. Para usar a função fibonacci, basta, no editor do MATLAB, escrever e executar a seguinte instrução:

```
fibonacci(inline('(x^2-sin(x))','x'),0,1,19)
```

Abaixo apresenta-se o resultado obtido.

| k | x1 | x2 | f(x1) | f(x2) | Novo intervalo |
|---|----------|----------|-----------|-----------|-------------------------|
| 1 | 0.381966 | 0.618034 | -0.226847 | -0.197468 | [0.000000 , 0.618034] |
| 2 | 0.236068 | 0.381966 | -0.178153 | -0.226847 | [0.236068 , 0.618034] |
| 3 | 0.381966 | 0.472136 | -0.226847 | -0.231877 | [0.381966 , 0.618034] |
| 4 | 0.472136 | 0.527864 | -0.231877 | -0.225049 | [0.381966 , 0.527864] |
| 5 | 0.437694 | 0.472136 | -0.232276 | -0.231877 | [0.381966 , 0.472136] |
| 6 | 0.416408 | 0.437694 | -0.231082 | -0.232276 | [0.416408 , 0.472136] |
| 7 | 0.437694 | 0.450850 | -0.232276 | -0.232465 | [0.437694 , 0.472136] |
| 8 | 0.450850 | 0.458980 | -0.232465 | -0.232371 | [0.437694 , 0.458980] |

| | | | | | |
|----|----------|----------|-----------|-----------|-------------------------|
| 9 | 0.445825 | 0.450850 | -0.232442 | -0.232465 | [0.445825 , 0.458980] |
| 10 | 0.450850 | 0.453956 | -0.232465 | -0.232448 | [0.445825 , 0.453956] |
| 11 | 0.448931 | 0.450850 | -0.232464 | -0.232465 | [0.448931 , 0.453956] |
| 12 | 0.450850 | 0.452037 | -0.232465 | -0.232461 | [0.448931 , 0.452037] |
| 13 | 0.450119 | 0.450850 | -0.232466 | -0.232465 | [0.448931 , 0.450850] |
| 14 | 0.449662 | 0.450119 | -0.232465 | -0.232466 | [0.449662 , 0.450850] |
| 15 | 0.450119 | 0.450393 | -0.232466 | -0.232466 | [0.449662 , 0.450393] |
| 16 | 0.449936 | 0.450119 | -0.232466 | -0.232466 | [0.449936 , 0.450393] |
| 17 | 0.450119 | 0.450210 | -0.232466 | -0.232466 | [0.450119 , 0.450393] |
| 18 | 0.450210 | 0.450301 | -0.232466 | -0.232466 | [0.450119 , 0.450301] |
| 19 | 0.450210 | 0.450256 | -0.232466 | -0.232466 | [0.450119 , 0.450210] |

O intervalo de incerteza final é [0.450119 , 0.450210]

Uma aproximação para o minimizante é $x^* = 0.450164$

Uma aproximação para o mínimo é $f(x^*) = -0.232466$

O método de Fibonacci tem algumas desvantagens práticas⁵. Por exemplo, o número de iterações tem que ser definido a priori e é necessário calcular números de Fibonacci. Para tentar tirar partido das suas vantagens e obviar algumas das suas desvantagens, surge o método da secção áurea. Este não “comprime” tanto o intervalo, mas a diferença é pouca — cerca de 17%, como prova [1, Antoniou e Lou, p. 94].

2.5 Método da Secção Áurea

O método da secção áurea⁶ é análogo ao método de Fibonacci, derivando de se considerar que λ_k é constante, e igual a λ , nas equações (2.6) e (2.7). Nesse pressuposto teremos então que os pontos intermédios x_1 e x_2 , na iteração k , ($k \in \mathbb{N}$), verificam as relações

$$x_1 - a_k = \lambda(b_k - a_k) \text{ e } b_k - x_2 = \lambda(b_k - a_k), \quad (2.15)$$

que são equivalentes a

$$x_1 = a_k + \lambda(b_k - a_k) \text{ e } x_2 = b_k - \lambda(b_k - a_k). \quad (2.16)$$

Deste modo, deixará de ser necessário conhecer o número n de iterações para determinar os pontos intermédios x_1 e x_2 . Aqui reside a grande diferença entre os métodos de Fibonacci e da secção áurea.

No final da iteração $k + 1$, a amplitude do intervalo de incerteza é

$$b_{k+1} - a_{k+1} = (1 - \lambda)(b_k - a_k), \quad (2.17)$$

⁵Na Secção 2.6 são discutidas as desvantagens do método de Fibonacci.

⁶Na literatura internacional este método é conhecido por *golden section method*.

sendo que, ao fim de n iterações, verifica-se

$$b_n - a_n = (1 - \lambda)^n (b_0 - a_0). \quad (2.18)$$

Ao fim de cada iteração a redução do intervalo de incerteza é, portanto, constante e igual a $1 - \lambda$.

Dada uma precisão δ para o intervalo de incerteza final, a condição

$$\begin{aligned} b_n - a_n &< \delta, \\ (1 - \lambda)^n (b_0 - a_0) &< \delta, \\ (1 - \lambda)^n &< \frac{\delta}{b_0 - a_0}, \end{aligned} \quad (2.19)$$

dá-nos o número n de iterações necessárias (e, portanto, $n + 1$ experiências) para alcançarmos a precisão desejada.

Reescrevendo a Igualdade (2.9), considerando que λ_k é constante e $\lambda_k = \lambda$, obtém-se a equação

$$\lambda = 1 - \frac{\lambda}{1 - \lambda}, \quad \text{com } 0 \leq \lambda \leq \frac{1}{2}. \quad (2.20)$$

A constante λ pode ser obtida resolvendo a condição (2.20). No intervalo $\left[0, \frac{1}{2}\right]$, tem-se

$$\lambda = 1 - \frac{\lambda}{1 - \lambda} \Leftrightarrow \lambda(1 - \lambda) = 1 - 2\lambda \Leftrightarrow \lambda^2 - 3\lambda + 1 = 0 \Leftrightarrow \lambda = \frac{3 - \sqrt{5}}{2} \approx 0.382. \quad (2.21)$$

Note-se que em virtude de $1 - \lambda = 1 - \frac{3 - \sqrt{5}}{2} = \frac{\sqrt{5} - 1}{2}$ estar contido no intervalo $]0, 1[$,

$$\lim_{n \rightarrow +\infty} (b_n - a_n) = \lim_{n \rightarrow +\infty} \left[\left(\frac{\sqrt{5} - 1}{2} \right)^n (b_0 - a_0) \right] = 0 \times (b_0 - a_0) = 0.$$

O algoritmo do método da secção áurea, obtém-se do algoritmo do método de Fibonacci considerando que a taxa de redução do intervalo de incerteza, de iteração para iteração, é a constante $1 - \lambda = \frac{\sqrt{5} - 1}{2}$. Neste caso não há nenhum inconveniente com a colocação dos pontos intermédios na última iteração, como ocorria no método de Fibonacci. Tal como neste último método, pretende-se em cada iteração, depois da primeira, determinar apenas uma nova experiência, reutilizando uma da iteração anterior. Naturalmente, a redução do intervalo de incerteza é feita tendo em conta a unimodalidade da função objetivo. O Algoritmo 2.4 permite operacionalizar o método da secção áurea.

Algoritmo 2.4: Método da Secção Áurea**Entrada:** Intervalo inicial de incerteza $[a, b]$, tolerância δ e função objetivo f .

```

 $k := 1$ 
 $\lambda = \frac{3 - \sqrt{5}}{2}$ 
 $x_1 := a + \lambda(b - a)$ 
 $x_2 := b - \lambda(b - a)$ 
enquanto  $b - a \geq \delta$  faça
  | se  $f(x_1) < f(x_2)$  então
  |   |  $b := x_2$ 
  |   |  $x_2 := x_1$ 
  |   |  $x_1 := a + \lambda(b - a)$ 
  | senão
  |   |  $a := x_1$ 
  |   |  $x_1 := x_2$ 
  |   |  $x_2 := b - \lambda(b - a)$ 
  | fim
  |  $k := k + 1$ 
fim

```

O intervalo de incerteza final é $[a, b]$ $x^* := \frac{a + b}{2}$ Mín := $f(x^*)$

Antes de terminarmos o estudo do método da secção áurea com a resolução de alguns problemas, vamos justificar o facto de o método se designar por secção áurea. Começemos por notar que

$$1 - \lambda = \frac{\sqrt{5} - 1}{2} \text{ e que } \frac{\lambda}{1 - \lambda} = \frac{\frac{3 - \sqrt{5}}{2}}{\frac{\sqrt{5} - 1}{2}} = \frac{3 - \sqrt{5}}{\sqrt{5} - 1} = \frac{\sqrt{5} - 1}{2}.$$

Portanto,

$$\frac{\lambda}{1 - \lambda} = \frac{1 - \lambda}{1}.$$

Ou seja, dividindo um intervalo em duas partes, de comprimentos λ e $1 - \lambda$, resulta que a razão entre a parte menor e a maior é igual à razão da maior para a soma das duas. Esta regra era conhecida pelos geómetras da Grécia Antiga como a secção áurea. Ao usar-se a regra da secção áurea num método de minimização este passou a assumir a designação desta regra.

Problema 2.10 Considere que a função definida por $f(x) = \frac{\sin \frac{1}{x}}{(x - 0.2)^2 + 0.1}$ é unimodal

no intervalo $[0.13, 0.4]$. Determine o mínimo de f em $[0.13, 0.4]$, usando o método da secção áurea. Localize o minimizante num intervalo de amplitude inferior a 0.1.

Resolução:

Ser-nos-á útil determinar o número de iterações necessárias, uma vez que não nos vamos socorrer de nenhuma ferramenta computacional para resolver este problema. Atendendo as igualdades (2.19) e (2.21) tem-se:

$$\left(\frac{\sqrt{5}-1}{2}\right)^n < \frac{0.1}{0.4-0.13},$$

$$n > \frac{\ln\left(\frac{10}{27}\right)}{\ln\left(\frac{\sqrt{5}-1}{2}\right)}.$$

Tomamos $n = 3$ por ser o menor inteiro que verifica a desigualdade anterior.

1.^a iteração

Como o intervalo de incerteza inicial é $[0.13, 0.4]$, $a = 0.13$ e $b = 0.4$.

$$x_1 = a + \frac{3-\sqrt{5}}{2}(b-a) = 0.13 + \frac{3-\sqrt{5}}{2} \times (0.4-0.13) \approx 0.23313,$$

$$x_2 = b - \frac{3-\sqrt{5}}{2}(b-a) = 0.4 - \frac{3-\sqrt{5}}{2} \times (0.4-0.13) \approx 0.29687,$$

$$f(x_1) = f(0.23313) = \frac{\sin \frac{1}{0.23313}}{(0.23313-0.2)^2 + 0.1} \approx -9.01981,$$

$$f(x_2) = f(0.29687) = \frac{\sin \frac{1}{0.29687}}{(0.29687-0.2)^2 + 0.1} \approx -2.05655.$$

Atendendo que $f(x_1) < f(x_2)$, então elimina-se o intervalo $]x_2, b] =]0.29687, 0.4]$. O novo intervalo de incerteza é, em consequência, $[a, x_2] = [0.13, 0.29687]$.

2.^a iteração

Como o intervalo de incerteza obtido no final da 1.^a iteração é $[0.13, 0.29687]$, $a = 0.13$ e $b \approx 0.29687$.

$$x_1 = a + \frac{3 - \sqrt{5}}{2}(b - a) = 0.13 + \frac{3 - \sqrt{5}}{2} \times (0.29687 - 0.13) \approx 0.19374,$$

$$x_2 = 0.23313 \text{ (reutilização do valor } x_1 \text{ da iteração anterior),}$$

$$f(x_1) = f(0.19374) = \frac{\sin \frac{1}{0.19374}}{(0.19374 - 0.2)^2 + 0.1} \approx -9.00437,$$

$$f(x_2) = f(0.23313) = \frac{\sin \frac{1}{0.23313}}{(0.23313 - 0.2)^2 + 0.1} \approx -9.01981.$$

Uma vez que $f(x_1) > f(x_2)$, eliminaremos o intervalo $[a, x_1[= [0.13, 0.19374[$. Logo, o novo intervalo de incerteza é $[x_1, b] = [0.19374 \dots, 0.29687 \dots]$.

3.^a iteração

Nesta iteração final, vamos considerar o intervalo de incerteza $[0.19374 \dots, 0.29687 \dots]$. Logo, $a = 0.19374$ e $b = 0.29687$.

$$x_1 = 0.23313 \text{ (reutilização do valor } x_2 \text{ da iteração anterior),}$$

$$x_2 = b - \frac{3 - \sqrt{5}}{2}(b - a) = 0.29687 - \frac{3 - \sqrt{5}}{2} \times (0.29687 - 0.19374) \approx 0.25748,$$

$$f(x_1) = f(0.19374) = \frac{\sin \frac{1}{0.19374}}{(0.19374 - 0.2)^2 + 0.1} \approx -9.00437,$$

$$f(x_2) = f(0.23313) = \frac{\sin \frac{1}{0.23313}}{(0.23313 - 0.2)^2 + 0.1} \approx -9.01981.$$

Uma vez que $f(x_1) < f(x_2)$, então elimina-se o intervalo $]x_2, b] =]0.25748, 0.29687]$. Assim, obtém-se o intervalo de incerteza final $[a, x_2] = [0.19374 \dots, 0.25748 \dots]$.

Uma aproximação para o mínimo é

$$f\left(\frac{0.19374 + 0.25748}{2}\right) = f(0.22561) = \frac{\sin \frac{1}{0.22561}}{(0.22561 - 0.2)^2 + 0.1} \approx -9.5480.$$

Problema 2.11 Utilize o MATLAB para escrever uma rotina que implemente o método da secção áurea.

Resolução:

Para resolver este problema, optamos por uma solução que apresente os cálculos intermédios, registando numa tabela o número da iteração, os pontos intermédios, as suas imagens e o intervalo de incerteza no fim de cada iteração. A solução proposta para responder à questão colocada designa-se por seccaoaurea e é apresentada a seguir.

```
function seccaoaurea(f,a,b,delta)
%Determina o mínimo de uma função unidimensional unimodal pelo
%método da secção áurea
%-----DADOS DE ENTRADA-----
%f=função objetivo
%a=limite inferior do intervalo inicial de incerteza
%b=limite superior do intervalo inicial de incerteza
%delta= precisão do intervalo de incerteza
%Sintaxe para f: f=inline('x*sin(x)','x')
%Exemplo: seccaoaurea(inline('x*sin(x)','x'),4,5,0.001)
%-----DADOS DE SAÍDA-----
%a=limite inferior do intervalo de incerteza final
%b=limite superior do intervalo de incerteza final
%minimizante x^*
%minimo Min
%----- INÍCIO -----
z=zeros(1,6);
k=0;
lambda = (3-sqrt(5))/2;
x1 = a+lambda*(b-a);
x2 = b-lambda*(b-a);
fprintf('-----\n');
fprintf(' k      x1      x2      f(x1)      f(x2)\n');
fprintf(' Novo intervalo\n');
fprintf('-----\n');
while (b-a>=delta)
    z(1)=x1;
    z(2)=x2;
    z(3)=f(x1);
    z(4)=f(x2);
    if (f(x1)<f(x2))
        b = x2;
        x2 = x1;
        x1 = a + lambda*(b-a);
    elseif(f(x1)>f(x2))
```

```
a = x1;
x1 = x2;
x2 = b-lambda*(b-a);
end
z(5)=a;
z(6)=b;
k=k+1;
fprintf('%2d\t %.5f\t %.5f\t %.5f\t %.5f\t',k, z(1), z(2), z(3), z(4));
fprintf('%c %.5f %c %.5f %c\n', '[' ,z(5), ',' ,z(6), ']' );
end
fprintf('\n');
fprintf('O intervalo de incerteza final é ');
fprintf('%c %.5f %c %.5f %c\n', '[' ,a, ',' ,b, ']' );
fprintf('Uma aproximação para o minimizante é x^*=% .5f\n', (a+b)/2);
fprintf('Uma aproximação para o mínimo é f(x^*)= %.5f\n', f((a+b)/2));
end
```

Problema 2.12 *Seja g a função definida, em \mathbb{R} , por $g(x) = x^2 + 4 \cos x$.*

Sabe-se que g é unimodal no intervalo $[1, 2]$. Determine um valor aproximado para o mínimo de g , localizando o minimizante num intervalo de amplitude inferior a 0.001. Utilize a rotina do MATLAB proposta no Problema 2.11.

Resolução:

Para usar a função seccaoaurea, basta no editor do MATLAB, escrever e executar a seguinte instrução:

```
seccaoaurea(inline('x^2+4*cos(x)', 'x'), 1, 2, 0.001)
```

Abaixo apresenta-se o resultado obtido.

| k | x1 | x2 | f(x1) | f(x2) | Novo intervalo |
|---|---------|---------|---------|---------|-----------------------|
| 1 | 1.38197 | 1.61803 | 2.66067 | 2.42915 | [1.38197 , 2.00000] |
| 2 | 1.61803 | 1.76393 | 2.42915 | 2.34371 | [1.61803 , 2.00000] |
| 3 | 1.76393 | 1.85410 | 2.34371 | 2.31957 | [1.76393 , 2.00000] |
| 4 | 1.85410 | 1.90983 | 2.31957 | 2.31715 | [1.85410 , 2.00000] |
| 5 | 1.90983 | 1.94427 | 2.31715 | 2.32078 | [1.85410 , 1.94427] |
| 6 | 1.88854 | 1.90983 | 2.31689 | 2.31715 | [1.85410 , 1.90983] |

| | | | | | |
|----|---------|---------|---------|---------|-----------------------|
| 7 | 1.87539 | 1.88854 | 2.31747 | 2.31689 | [1.87539 , 1.90983] |
| 8 | 1.88854 | 1.89667 | 2.31689 | 2.31681 | [1.88854 , 1.90983] |
| 9 | 1.89667 | 1.90170 | 2.31681 | 2.31687 | [1.88854 , 1.90170] |
| 10 | 1.89357 | 1.89667 | 2.31681 | 2.31681 | [1.89357 , 1.90170] |
| 11 | 1.89667 | 1.89859 | 2.31681 | 2.31682 | [1.89357 , 1.89859] |
| 12 | 1.89549 | 1.89667 | 2.31681 | 2.31681 | [1.89357 , 1.89667] |
| 13 | 1.89476 | 1.89549 | 2.31681 | 2.31681 | [1.89476 , 1.89667] |
| 14 | 1.89549 | 1.89594 | 2.31681 | 2.31681 | [1.89476 , 1.89594] |
| 15 | 1.89521 | 1.89549 | 2.31681 | 2.31681 | [1.89521 , 1.89594] |

O intervalo de incerteza final é [1.89521 , 1.89594]

Uma aproximação para o minimizante é $x^* = 1.89557$

Uma aproximação para o mínimo é $f(x^*) = 2.31681$

2.6 Comparação de Métodos de Eliminação

A comparação que aqui será feita é especialmente importante quando a função objetivo é complicada e demorada de avaliar. Neste caso, poupar experiências é fundamental para a eficiência computacional dos algoritmos.

Consideremos que A_n representa a amplitude do intervalo de incerteza após n experiências e que A_0 representa a amplitude do intervalo de incerteza inicial.

Para cada um dos métodos estudados, na tabela 2.2, apresenta-se, na segunda coluna, o número n de experiências em função do número k de iterações ($k \in \mathbb{N}$ e $n \in \mathbb{N}$). Na terceira coluna, está a razão entre a amplitude do intervalo de incerteza ao fim de k iterações e a amplitude do intervalo de incerteza inicial. Na última coluna, está representada a razão entre a amplitude do intervalo de incerteza ao fim de n experiências e a amplitude do intervalo de incerteza inicial, $\frac{A_n}{A_0}$. Esta razão mede a eficiência de um método de eliminação.

Os valores de $\frac{A_n}{A_0}$ alcançados para vários valores do número n de experiências são comparados nas figuras 2.15 (pág. 76) e 2.16 (pág. 79). Note-se que o método de busca dicotômica só admite experiências em número par e o método de busca por bissecção, por sua vez, apenas admite experiências em número ímpar maior 1. No caso do método de busca dicotômica, considerou-se $A_0 = 1$ e a constante de diferenciação $\varepsilon = 0.001$.

Nas figuras 2.15 (pág. 76) e 2.16 (pág. 79) são comparados os métodos de eliminação intervalar estudados, do ponto de vista da sua eficiência. No primeiro caso foi usada uma sequência de números pares de experiências e no segundo números ímpares. Os gráficos foram obtidos usando uma escala logarítmica de base 10 no eixo das ordenadas e uma escala linear no eixo das abcissas.

| Método | Número n de experiências para k iterações ($k \in \mathbb{N}$) | Taxa de redução do intervalo de incerteza em função do número k de iterações, $\frac{A_k}{A_0}$ | Taxa de redução do intervalo de incerteza em função do número n de experiências, $\frac{A_n}{A_0}$ |
|----------------------------|--|---|--|
| Busca Dicotômica | $2k$ | $\frac{1}{2^k} + 0.002 \left(1 - \frac{1}{2^k}\right)$ $\varepsilon = 0.001; A_0 = 1$ | $\frac{1}{2^{0.5n}} + 0.002 \left(1 - \frac{1}{2^{0.5n}}\right)$ $\varepsilon = 0.001; A_0 = 1$ |
| Busca por Bissecção | $2k + 1$ | $\left(\frac{1}{2}\right)^k$ | $\left(\frac{1}{2}\right)^{\frac{n-1}{2}}$ |
| Fibonacci | $k + 1$ | $\frac{1}{F_{k+1}}$ | $\frac{1}{F_n}$ |
| Secção Áurea | $k + 1$ | $\left(\frac{\sqrt{5}-1}{2}\right)^k$ | $\left(\frac{\sqrt{5}-1}{2}\right)^{n-1}$ |

Tabela 2.2: Relação entre o número de iterações e o número de experiências. Taxa de redução do intervalo de incerteza em função do número de iterações ou do número de experiências.

As figuras 2.15 (pág. 76) e 2.16 (pág. 79) foram geradas com recurso à linguagem *R*. No caso da Figura 2.15 (pág. 76) foi usado o seguinte código:

```
x<-seq(2,54,by=2)#número de experiências (par)
S<-sqrt(5)
K<-2^(0.5*x)
Fib<-(((1+S)/2)^(x+1)-((1-S)/2)^(x+1))/S#números de Fibonacci
plot(x,1/K+2*0.001*(1-1/K),col="yellow",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5,xlab=
"Número de experiências",ylab="Taxa de redução do intervalo de incerteza")
lines(x,1/Fib,col="green",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5)#fibonacci
#lines(x,0.5^((x-1)/2),col="blue",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5)#bissecção
lines(x,((S-1)/2)^(x-1),col="red",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5)#áurea
legend(40,1.2,c("Dicotômica","Fibonacci","Sec. áurea"),col=c("yellow","green","red"),lwd=c(5,5,5))#legenda
```

A Figura 2.16 (pág. 79) foi obtida com o seguinte código:

```
x<-seq(3,54,by=2)#número de experiências (ímpar)
S<-sqrt(5)
```

```

K<-2^(0.5*x)
Fib<-(((1+S)/2)^(x+1)-((1-S)/2)^(x+1))/S#números de Fibonacci
plot(x,0.5^((x-1)/2),col="blue",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5,
xlab="Número de experiências",ylab="Taxa de redução do intervalo de incerteza")
lines(x,1/Fib,col="green",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5)#fibonacci
lines(x,((S-1)/2)^(x-1),col="red",type="o",log="y",pch=20,xlim=c(0,60),ylim=c(10^-12,2),lwd=1.5)#áurea
legend(40,1.2,c("Bisseção","Fibonacci","Sec. áurea"),col=c("blue","green","red"),lwd=c(5,5,5))#legenda

```

Analisando as figuras 2.15 (pág. 76) e 2.16 (pág. 79), constata-se que, na redução do intervalo de incerteza, o método de Fibonacci é o mais eficiente, seguido pelo método da secção áurea. O método de Fibonacci é o que contrai mais o intervalo de incerteza para um dado número de experiências. Se só contarmos o número de experiências e se só cuidarmos da contração do intervalo inicial, este método é o melhor dos métodos analisados neste capítulo. A solução⁷ do Problema (A.2) (pág. 145) garante que o método de Fibonacci é um método de otimização ótimo.

As figuras 2.15 (pág. 76) e 2.16 (pág. 79) mostram ainda que:

- o método de busca dicotômica não funciona a partir de certa altura;
- o logaritmo decimal da taxa de redução do intervalo de incerteza no método de busca por bisseção é constante e igual a $\frac{\sqrt{2}}{2}$;
- o logaritmo decimal da taxa de redução do intervalo de incerteza no método de Fibonacci e no método da secção áurea é cerca de $0.618\dots$.

Quando o número de iterações é grande, no método de Fibonacci tem-se que $\lambda_k \approx \frac{3 - \sqrt{5}}{2}$. Este valor é o valor da constante λ usada no método da secção áurea. A proposição seguinte relaciona os dois métodos anteriores.

Proposição 2.1 *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci. Então:*

(i) *a sucessão de termo geral $G_n = \frac{F_n}{F_{n+1}}$ é convergente;*

(ii) $\lim_{n \rightarrow +\infty} \frac{F_n}{F_{n+1}} = \frac{\sqrt{5} - 1}{2}.$

⁷Ver Teorema A.1, pág. 151.

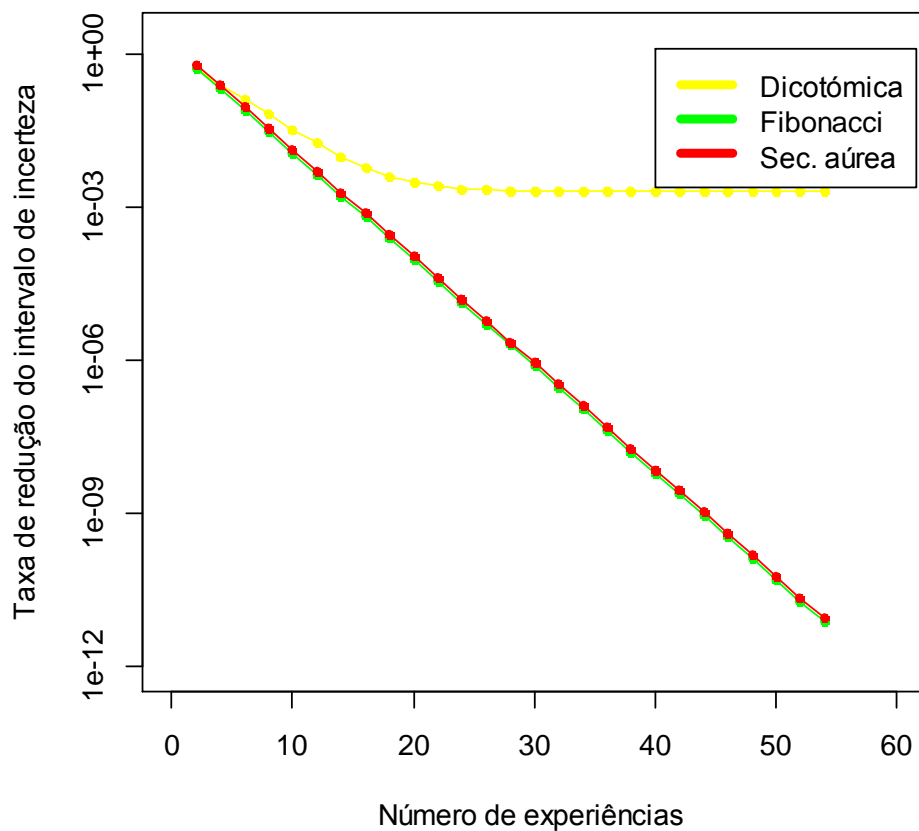


Figura 2.15: Comparação dos métodos de busca dicotômica, de Fibonacci e da secção áurea — número par de experiências.

Prova: Usando a Fórmula de Binet (Proposição A.7, pág. 153), obtém-se

$$G_n = \frac{F_n}{F_{n+1}},$$

$$G_n = \frac{\frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^{n+1} - \left(\frac{1 - \sqrt{5}}{2} \right)^{n+1} \right]}{\frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^{n+2} - \left(\frac{1 - \sqrt{5}}{2} \right)^{n+2} \right]},$$

$$G_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}}{\left(\frac{1+\sqrt{5}}{2}\right)^{n+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+2}},$$

$$G_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}}{\left(\frac{1+\sqrt{5}}{2}\right)^{n+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+2}},$$

$$G_n = \frac{\frac{\sqrt{5}-1}{2} - \left(\frac{\sqrt{5}-3}{2}\right)^n (2-\sqrt{5})}{1 - \left(\frac{\sqrt{5}-3}{2}\right)^n \left(\frac{7-3\sqrt{5}}{2}\right)}.$$

Uma vez que a sucessão $(G_n)_{n \in \mathbb{N}_0}$ é definida pelo quociente de sucessões convergentes, por serem constantes ou exponenciais de base no intervalo $] -1, 1[$, conclui-se que esta é convergente. Fica, assim, provada a afirmação (i).

$$\lim_{n \rightarrow +\infty} G_n = \lim_{n \rightarrow +\infty} \frac{\frac{\sqrt{5}-1}{2} - \left(\frac{\sqrt{5}-3}{2}\right)^n (2-\sqrt{5})}{1 - \left(\frac{\sqrt{5}-3}{2}\right)^n \left(\frac{7-3\sqrt{5}}{2}\right)},$$

$$\lim_{n \rightarrow +\infty} G_n = \frac{\frac{\sqrt{5}-1}{2} - 0 \times (2-\sqrt{5})}{1 - 0 \times \left(\frac{7-3\sqrt{5}}{2}\right)},$$

$$\lim_{n \rightarrow +\infty} G_n = \frac{\sqrt{5}-1}{2}.$$

O que mostra que (ii) se verifica e permite concluir a prova. □

Na Secção 2.4 vimos que no método de Fibonacci, na iteração k , o fator de redução do intervalo de incerteza é $1 - \lambda_k = \frac{F_{n-k+1}}{F_{n-k+2}}$ (quando se pretende realizar n iterações). Por outro lado, no método da secção áurea (ver Secção 2.5) concluímos que, de iteração para iteração, o fator de redução do intervalo de incerteza é $1 - \lambda = \frac{\sqrt{5}-1}{2}$. Decorre, portanto, da Proposição 2.1 que para um número n muito grande de iterações se tem $1 - \lambda_k \approx 1 - \lambda = \frac{\sqrt{5}-1}{2}$. Ou seja, a taxa de contração do intervalo de incerteza é sensivelmente a mesma nos dois métodos. Este argumento dá vantagem ao método da secção áurea, dado que este não requer o cálculo de números de Fibonacci.

O método da secção áurea é o mais utilizado na prática, apesar de o método de Fibonacci ser mais eficiente. Este facto tem a ver com os seguintes fatores:

1. Se o número de iterações n é elevado, a contração obtida no intervalo de incerteza é quase igual à do método de Fibonacci;
2. Não é necessário calcular números de Fibonacci. Fixa-se a constante $0.381966 \dots$;
3. Ao contrário do método de Fibonacci, não é necessário fixar, a priori, o número de iterações;
4. O método da secção áurea para quando quisermos, mesmo usando critérios de paragem decididos no decorrer do processo. É possível parar e, se não estivermos satisfeitos, podemos continuar mais algumas iterações;
5. A última iteração no método da secção áurea é natural, ao contrário do método de Fibonacci, onde, na última iteração, a escolha de θ pode não ser fácil de automatizar no caso geral;
6. Se o número de iterações for muito grande, o método de Fibonacci começa a colocar experiências muito próximas e pode haver problemas. Caso haja erros de arredondamento, pode acontecer $x_1 > x_2$. Por outro lado, estando x_1 muito próximo de x_2 , ter-se-á $f(x_1) \approx f(x_2)$ e um pequeno erro no cálculo de $f(x_1)$ e/ou $f(x_2)$ pode levar a eliminar um intervalo que contenha o minimizante. Por estas razões, o método da secção áurea é mais robusto, no sentido em que falha menos;
7. O método da secção áurea é mais fácil de perceber e de programar.

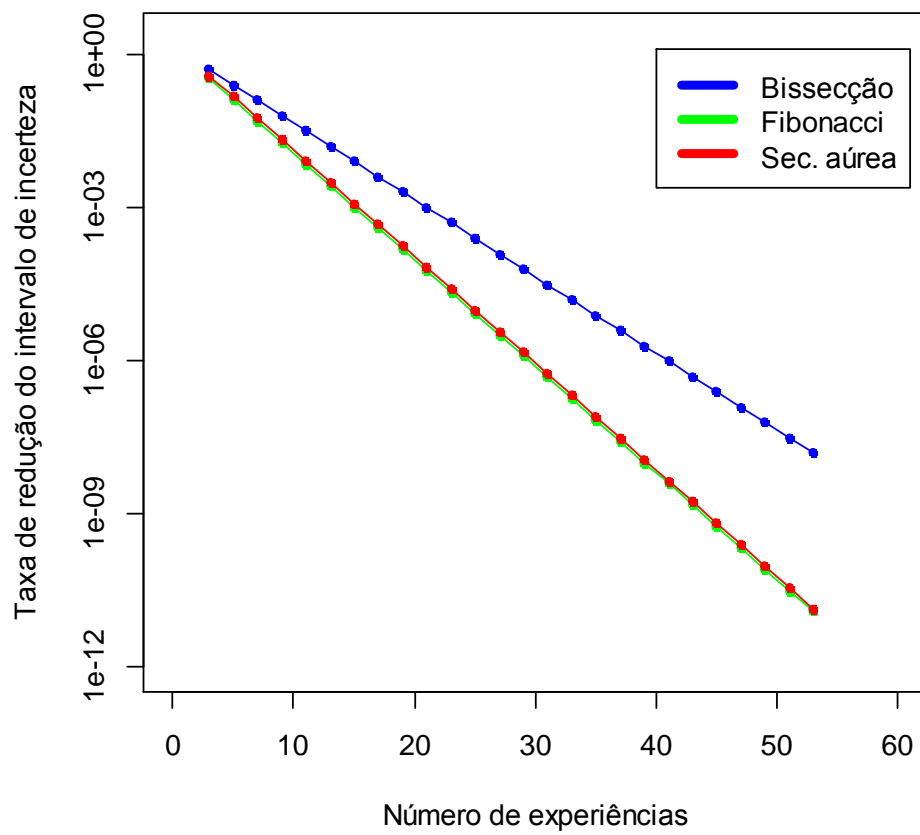


Figura 2.16: Comparação dos métodos de busca por bissecção, de Fibonacci e da secção áurea — número ímpar de experiências.

Capítulo 3

Métodos de Aproximação Polinomial

No Capítulo 2 estudámos métodos intervalares de eliminação aplicados ao problema

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a} & x \in \mathcal{A}, \quad \text{com } \mathcal{A} = [a, b] \text{ e } f \text{ unimodal.} \end{array}$$

Restringiremos o estudo ao caso em que f é contínua em $[a, b]$, o mesmo sucedendo com f' .

Neste capítulo, vamos estudar métodos que resolvem o problema atrás referido, mas que aproximam f por um polinómio $p(x)$. Isto é,

$$p(x) \approx f(x), \quad \forall x \in [a, b].$$

O polinómio $p(x)$ será de grau 2 ou 3. Deste modo, a equação $p'(x) = 0$ será uma equação de resolução imediata, por ser uma equação linear ou quadrática. A sua resolução permite encontrar um minimizante local \bar{x} de $p(x)$ em $[a, b]$. \bar{x} é uma aproximação do minimizante x^* de f em $[a, b]$. Depois de calculado \bar{x} , o intervalo de incerteza $[a, b]$ é reduzido usando a hipótese de unimodalidade de f em $[a, b]$. Para o efeito, calcula-se $f(\bar{x})$ e $f(\alpha)$, sendo $\alpha \neq \bar{x}$ um ponto de $]a, b[$. O processo descrito será repetido até que se encontre uma aproximação satisfatória para x^* .

Diremos que um polinómio $p(x)$ *interpola* uma função f num ponto x_0 se $p(x_0) = f(x_0)$. Por outro lado, se além de $p(x)$ interpolar uma função f num ponto x_0 , se verificar $p'(x_0) = f'(x_0)$, diremos que $p(x)$ *oscula* f em x_0 .

Os métodos interpoladores de aproximação diferem apenas na forma de definir o polinómio $p(x)$. O polinómio interpolará f em dois ou mais pontos, osculando a função, eventualmente, em um dos pontos de interpolação.

O desenvolvimento deste capítulo será baseado nos autores [1, Antoniou e Lou], [4, Chapra e Canale], [7, GAVE], [11, Sá Esteves], [18, Higham], [19, Kiusalaas], [20, Knight], [21, Lima] e [26, Rao].

3.1 Método de Interpolação Quadrática

3.1.1 Interpolação em Três Pontos

No método de interpolação quadrática o polinómio aproximante $p(x)$ é um polinómio do 2.^o grau. Portanto,

$$p(x) = a_2x^2 + a_1x + a_0, \quad (3.1)$$

onde a_0 , a_1 e a_2 são constantes, sendo $a_2 \neq 0$.

Consideremos uma função f unimodal no intervalo $[x_1, x_3]$ e $x_2 \in]x_1, x_3[$. Adicionalmente, suponhamos que $p(x)$ interpola f em x_1 , x_2 e x_3 , sendo conhecidos $f(x_1) = f_1$, $f(x_2) = f_2$ e $f(x_3) = f_3$. Por conseguinte, as constantes a_0 , a_1 e a_2 podem ser obtidas resolvendo o sistema de equações

$$\begin{cases} a_2x_1^2 + a_1x_1 + a_0 = f_1 \\ a_2x_2^2 + a_1x_2 + a_0 = f_2 \\ a_2x_3^2 + a_1x_3 + a_0 = f_3 \end{cases} \quad (3.2)$$

Do sistema (3.2) deduz-se a forma matricial

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \times \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}. \quad (3.3)$$

Uma vez que os elementos de cada linha da matriz do sistema estão em progressão geométrica, esta é uma matriz de Vandermonde, sendo determinada pelo vetor (x_1, x_2, x_3) . Por outro lado, como $x_1 < x_2 < x_3$, a matriz é não singular¹, sendo por isso o seu determinante diferente de zero. Portanto, a matriz é invertível, pelo que a solução do sistema é única, ou seja, os valores de a_0 , a_1 e a_2 determinam-se de forma única. Portanto, $p(x)$ é uma aproximação de $f(x)$ em $[x_1, x_3]$, como se mostra na Figura 3.1. Como se pode ver, os minimizantes \bar{x} de $p(x)$ e x^* de f estão próximos e se $f(x)$ puder ser aproximada por uma função quadrática, então $x^* \approx \bar{x}$. No caso de f ser uma função quadrática, ter-se-á $f(x) = p(x)$ e, consequentemente, $x^* = \bar{x}$.

Vejamos como determinar o minimizante \bar{x} de $p(x)$. Antes de mais, importa referir que uma vez que f é unimodal em $[x_1, x_3]$, então $p(x)$ representa uma função convexa e, como $a_2 \neq 0$, tem-se que $a_2 > 0$. Em virtude de $p(x)$ ser diferenciável em $]x_1, x_3[$ e $\bar{x} \in]x_1, x_3[$, então $p'(\bar{x}) = 0$. Como

$$p'(x) = 2a_2x + a_1,$$

¹No Apêndice B, prova-se que, sob determinadas condições, a matriz de Vandermonde tem determinante não nulo.

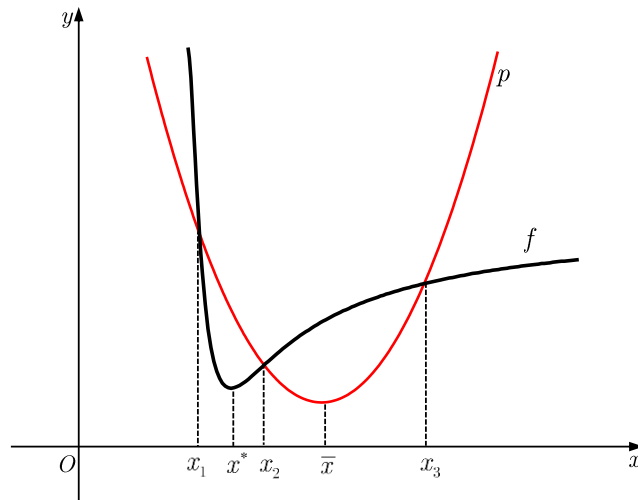


Figura 3.1: Interpolação em três pontos.

então tem-se

$$p'(\bar{x}) = 0,$$

$$2a_2\bar{x} + a_1 = 0,$$

$$\bar{x} = -\frac{a_1}{2a_2}. \quad (3.4)$$

A solução do sistema de equações (3.2) pode ser obtida resolvendo a Equação (3.3). Atendendo à Igualdade (3.4), podemos, com recurso ao **MATLAB**, obter a solução do sistema e uma expressão para \bar{x} em função de x_1, x_2, x_3, f_1, f_2 e f_3 . Para tal basta executar no seu editor o seguinte código:

```
clear global
syms x1 x2 x3 f1 f2 f3 a0 a1 a2
M=[x1^2 x1 1; x2^2 x2 1; x3^2 x3 1]
F=[f1; f2; f3];
A=inv(M)*F;
a2=A(1);
a1=A(2);
a0=A(3);
xbar=simplify(-a1/(2*a2))
```

O resultado que se obtém permite calcular \bar{x} e é

```
xbar =
(f1*x2^2 - f2*x1^2 - f1*x3^2 + f3*x1^2 + f2*x3^2 - f3*x2^2)/
(2*f1*x2 - 2*f2*x1 - 2*f1*x3 + 2*f3*x1 + 2*f2*x3 - 2*f3*x2).
```

Vamos reescrever a expressão obtida, por aplicação do código atrás referido, para uma forma mais legível. Essa forma é

$$\bar{x} = \frac{f_1x_2^2 - f_2x_1^2 - f_1x_3^2 + f_3x_1^2 + f_2x_3^2 - f_3x_2^2}{2f_1x_2 - 2f_2x_1 - 2f_1x_3 + 2f_3x_1 + 2f_2x_3 - 2f_3x_2}. \quad (3.5)$$

A Igualdade (3.5) é equivalente a

$$\bar{x} = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]}. \quad (3.6)$$

Fazendo $x^* \approx \bar{x}$, atendendo à Igualdade (3.6), conclui-se a 1.^a iteração do método de interpolação quadrática. Se pretendermos uma aproximação melhor, torna-se necessário executar mais iterações, reduzindo o intervalo de incerteza $[x_1, x_3]$ atendendo à hipótese de unimodalidade de f . O procedimento é o que usamos para os métodos estudados no Capítulo 2 e melhor descrito na Secção 2.1. Note-se que dispomos de dois pontos no interior do intervalo $[x_1, x_3]$: os pontos x_2 e \bar{x} (que são distintos no caso geral).

No caso de a função f não poder ser representada com precisão por um polinómio do segundo grau, a estratégia apropriada passa por tentar reduzir o intervalo de incerteza como se procedeu com os métodos intervalares de eliminação, no Capítulo 2. Este objetivo pode ser alcançado rejeitando ou x_1 ou x_3 e usando os dois pontos restantes juntamente com o ponto \bar{x} para nova interpolação.

Ao fim de um certo número de iterações, os três pontos estarão numa vizinhança de x^* . Consequentemente, o polinómio do segundo grau correspondente será uma representação precisa de $f(x)$, atendendo à série de Taylor². Deste modo, x^* pode ser determinado com a precisão desejada aplicando o algoritmo 3.1 (pág. 84), o qual se baseia nos princípios acima expostos.

No *Passo 4* do algoritmo 3.1 (pág. 84), o intervalo de incerteza $[x_1, x_3]$ é reduzido criteriosamente para $[x_1, x_2]$ ou $[\bar{x}, x_3]$, se $x_1 < \bar{x} < x_2$, ou para $[x_2, x_3]$ ou $[x_1, \bar{x}]$, se $x_2 < \bar{x} < x_3$, usando a condição de unimodalidade da função objetivo. O algoritmo implica uma avaliação da função objetivo por iteração, excepto na 1.^a iteração em que são necessárias três avaliações da função objetivo.

Em virtude do elevado número de cálculos repetitivos, decorrentes da utilização do algoritmo 3.1 (pág. 84), a sua implementação computacional é muito útil, razão pela qual usaremos a função *interpolacao_quadratca*, escrita em linguagem MATLAB e adaptada da

²Este argumento é válido pelo menos para funções de classe \mathcal{C}^2 .

Algoritmo 3.1: Método de Interpolação Quadrática

Passo 1

Entrada: Intervalo inicial de incerteza $[x_1, x_3]$, tolerância ε , função objetivo f e $\bar{x}_0 = 10^{308}$.

Passo 2

Calcular:
 $x_2 = \frac{x_1 + x_3}{2}$; $f_1 = f(x_1)$; $f_2 = f(x_2)$; $f_3 = f(x_3)$

Passo 3

Calcular:
$$\bar{x} = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]}$$

$\bar{f} = f(\bar{x})$
se $|\bar{x} - \bar{x}_0| < \varepsilon$ **então**
 $x^* = \bar{x}$
 Stop

Passo 4

se $x_1 < \bar{x} < x_2$ **então**
 se $\bar{f} \leq f_2$ **então**
 $x_3 = x_2$
 $f_3 = f_2$
 $x_2 = \bar{x}$
 $f_2 = \bar{f}$
 senão se $\bar{f} > f_2$ **então**
 $x_1 = \bar{x}$; $f_1 = \bar{f}$
 fim
senão se $x_2 < \bar{x} < x_3$ **então**
 se $\bar{f} \leq f_2$ **então**
 $x_1 = x_2$
 $f_1 = f_2$
 $x_2 = \bar{x}$
 $f_2 = \bar{f}$
 senão se $\bar{f} > f_2$ **então**
 $x_3 = \bar{x}$; $f_3 = \bar{f}$
 fim
fim
 $x^* \approx \bar{x}$

Voltar ao **Passo 3**.

função *quad_inter* proposta por [1, Antoniou e Lou]. A função requer quatro parâmetros: a função objetivo, os limites do intervalo de incerteza e uma tolerância de otimização.

Como resultado, a função apresenta uma aproximação para o minimizante e outra para o mínimo, bem como o número de iterações realizadas. Eis a função *interpolacao_quadratica*:

```
% Programa: interpolacao_quadratica.m
% Título: Método de Interpolação Quadrática
% Descrição: Implementação do Método de Interpolação Quadrática.
% Entrada:
%     fname: função objetiva
%     [xL1,xU1]: intervalo inicial de incerteza
%     epsi: tolerância de otimização
% Saída:
%     xs: minimizante
%     fs: mínimo da função objetivo
%     k: número de iterações
% Exemplo:
% Determinar o mínimo de
%     f(x)=x*sin(x)
% sabendo que 4<= x <= 5.
% Solução:
% Executar o comando
%     interpolacao_quadratica(inline('x*sin(x)','x'),4,5,0.03)
%-----
function interpolacao_quadratica(fname,x1,x3,epsi)
disp(' ');
disp('PROGRAMA interpolacao_quadratica.m')
x0bar = 1e308;
% Realização da 1.ª iteração
x2 = 0.5*(x1 + x3);
f1 = feval(fname,x1);
f2 = feval(fname,x2);
f3 = feval(fname,x3);
z1 = (x2 - x3)*f1;
z2 = (x3 - x1)*f2;
z3 = (x1 - x2)*f3;
z4 = (x2 + x3)*z1+(x3 + x1)*z2+(x1 + x2)*z3;
xbar = z4/(2*(z1 + z2 + z3));
fbar = feval(fname,xbar);
d = abs(x0bar - xbar);
k = 1;
% Realização da busca baseada na interpolação quadrática
while d >= epsi,
    if x1 < xbar & xbar < x2,
        if fbar <= f2,
```

```
        x3 = x2;
        f3 = f2;
        x2 = xbar;
        f2 = fbar;
    else
        x1 = xbar;
        f1 = fbar;
    end
elseif x2 < xbar & xbar < x3,
    if fbar <= f2,
        x1 = x2;
        f1 = f2;
        x2 = xbar;
        f2 = fbar;
    else
        x3 = xbar;
        f3 = fbar;
    end
end
x0bar=xbar;
z1 = (x2 - x3)*f1;
z2 = (x3 - x1)*f2;
z3 = (x1 - x2)*f3;
z4 = (x2 + x3)*z1 + (x3 + x1)*z2 + (x1 + x2)*z3;
xbar = z4/(2*(z1 + z2 + z3));
fbar = feval(fname,xbar);
d = abs(x0bar - xbar);
k = k+1;
end
% Apresentação de resultados
xs = xbar;
fs = feval(fname,xbar);
fprintf('Valor aproximado para o MINIMIZANTE: ');
fprintf(' %.6f\n', xs);
fprintf('Valor aproximado para o MÍNIMO: %.6f\n',fs);
fprintf('Número de ITERAÇÕES realizadas: %d\n',k);
end
```

Se, na Igualdade (3.6), assumirmos que os pontos x_1 , x_2 e x_3 estão igualmente espaçados,

obtem-se, para $\delta > 0$, $x_1 = x_2 - \delta$ e $x_3 = x_2 + \delta$,

$$\begin{aligned}\bar{x} &= \frac{(x_2 - x_3)(x_2 + x_3)f_1 + (x_3 - x_1)(x_3 + x_1)f_2 + (x_1 - x_2)(x_1 + x_2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]}, \\ \bar{x} &= \frac{-\delta(x_2 - x_1 + \frac{x_1 + x_3}{2} \times 2)f_1 + 2\delta\frac{x_1 + x_3}{2} \times 2f_2 - \delta(x_1 - x_2 + x_2 + x_2)f_3}{2(-\delta f_1 + 2\delta f_2 - \delta f_3)}, \\ \bar{x} &= \frac{-\delta(\delta + 2x_2)f_1 + 2\delta \times 2x_2f_2 - \delta(-\delta + 2x_2)f_3}{-2\delta(f_1 - 2f_2 + f_3)}, \\ \bar{x} &= \frac{(\delta + 2x_2)f_1 - 4x_2f_2 + (-\delta + 2x_2)f_3}{2(f_1 - 2f_2 + f_3)}, \\ \bar{x} &= \frac{\delta f_1 + 2x_2f_1 - 4x_2f_2 - \delta f_3 + 2x_2f_3}{2(f_1 - 2f_2 + f_3)}, \\ \bar{x} &= \frac{2x_2(f_1 - 2f_2 + f_3) + (f_1 - f_3)\delta}{2(f_1 - 2f_2 + f_3)}, \\ \bar{x} &= x_2 + \frac{(f_1 - f_3)\delta}{2(f_1 - 2f_2 + f_3)}.\end{aligned}\tag{3.7}$$

A Fórmula (3.7) envolve muito menos cálculos do que a Igualdade (3.6), razão pela qual deve ser usada quando tivermos a garantia de que os pontos x_1 , x_2 e x_3 estão igualmente espaçados.

Para melhor entendermos o algoritmo 3.1, vamos aplicá-lo à resolução de um problema concreto, executando algumas iterações. Vamos também, a seguir, ilustrar a aplicação do programa *interpolacao_quadratica.m*.

Problema 3.1 *Pretende-se ligar uma fábrica, F , a uma central de tratamento de resíduos, C , por meio de uma conduta, conforme a Figura 3.2. A conduta deve seguir ao longo de um muro até um certo ponto, B , e daí deve seguir em linha reta até à central de tratamento. Designou-se por A o ponto do muro mais próximo da central de tratamento.*

A distância da fábrica ao ponto A é de 4 Km, e a distância deste ponto à central é de 2 Km.

Designou-se por x a distância, em Km, entre A e B . O preço da colocação da conduta é de 15000.00 € por Km, ao longo do muro, e 25000.00 € por Km, do muro à central de tratamento.

Determine o valor de x ($x \in [0.1, 3.9]$) para o qual o preço da colocação da conduta é mínimo. Realize três iterações do método de interpolação quadrática.

Resolução:

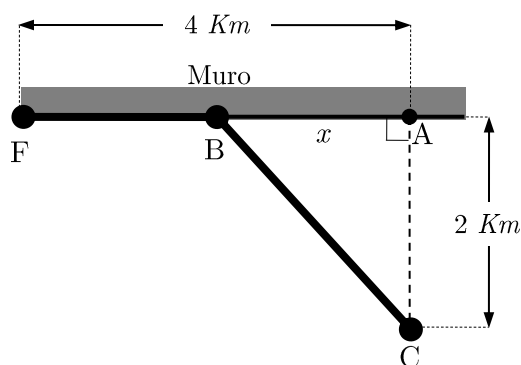


Figura 3.2: Esquematização do Problema 3.1.

Vamos começar por definir a função objetivo e , em seguida, vamos aplicar o algoritmo do método de interpolação quadrática.

Aplicando o Teorema de Pitágoras ao triângulo $[ABC]$ obtém-se

$$\overline{BC}^2 = \overline{AC}^2 + x^2,$$

$$\overline{BC}^2 = 2^2 + x^2,$$

$$\overline{BC} = \sqrt{x^2 + 4}.$$

A função objetivo representa o preço da colocação da conduta. Seja f a função objetivo. Como $\overline{FB} = 4 - x$ e o preço da colocação da conduta é dado, em milhares de euros, por $25 \times \overline{BC} + 15 \times \overline{FB}$, então

$$f(x) = 15(4 - x) + 25\sqrt{x^2 + 4}, \text{ com } x \in [0.1, 3.9].$$

A função f é unimodal no intervalo $[0.1, 3.9]$.

1.^a iteração

Passo 1

Como o intervalo de incerteza inicial é $[0.1, 3.9]$, $x_1 = 0.1$ e $x_3 = 3.9$.

Passo 2

$$x_2 = \frac{x_1 + x_3}{2} = \frac{0.1 + 3.9}{2} = 2,$$

$$f_1 = f(0.1) = 15(4 - 0.1) + 25\sqrt{0.1^2 + 4} \approx 108.5625,$$

$$f_2 = f(2) = 15(4 - 2) + 25\sqrt{2^2 + 4} \approx 100.7107,$$

$$f_3 = f(3.9) = 15(4 - 3.9) + 25\sqrt{3.9^2 + 4} \approx 111.0730.$$

Passo 3

$$\bar{x} = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]},$$

$$\bar{x} \approx \frac{(2^2 - 3.9^2) \times 108.5625 + (3.9^2 - 0.1^2) \times 100.7107 + (0.1^2 - 2^2) \times 111.0730}{2[(2 - 3.9) \times 108.5625 + (3.9 - 0.1) \times 100.7107 + (0.1 - 2) \times 111.0730]},$$

$$\bar{x} \approx 1.8691,$$

$$\bar{f} = f(1.8691) = 15(4 - 1.8691) + 25\sqrt{1.8691^2 + 4} \approx 100.3993.$$

Passo 4

Como $x_1 < \bar{x} < x_2$ e $\bar{f} < f_2$, então:

- $x_3 = x_2 = 2$,
- $f_3 = f_2 \approx 100.7107$,
- $x_2 = \bar{x} \approx 1.8691$,
- $f_2 = \bar{f} \approx 100.3993$,
- os valores de x_1 e f_1 mantêm-se ($x_1 = 0.1$ e $f_1 \approx 108.5625$).

Uma aproximação para o minimizante $x^* \approx \bar{x} \approx 1.8691$. Deste modo, concluímos a primeira iteração.

2.^a iteração

No final da iteração anterior, obtivemos $x_1 = 0.1$, $x_2 = 1.8691$, $x_3 = 2$, $f_1 \approx 108.5625$,

$f_2 \approx 100.3993$ e $f_3 \approx 100.7107$.

Passo 3

$$\bar{x} = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]},$$

$$\bar{x} \approx \frac{(1.8691^2 - 2^2) \times 108.5625 + (2^2 - 0.1^2) \times 100.3993 + (0.1^2 - 1.8691^2) \times 100.7107}{2[(1.8691 - 2) \times 108.5625 + (2 - 0.1) \times 100.3993 + (0.1 - 1.8691) \times 100.7107]},$$

$$\bar{x} \approx 1.6114,$$

$$\bar{f} = f(1.6114) = 15(4 - 1.6114) + 25\sqrt{1.6114^2 + 4} \approx 100.0387.$$

Passo 4

Como $x_1 < \bar{x} < x_2$ e $\bar{f} < f_2$, então:

- $x_3 = x_2 \approx 1.8691$,
- $f_3 = f_2 \approx 100.3993$,
- $x_2 = \bar{x} \approx 1.6114$,
- $f_2 = \bar{f} \approx 100.0387$,
- os valores de x_1 e f_1 mantêm-se ($x_1 = 0.1$ e $f_1 \approx 108.5625$).

Uma aproximação para o minimizante $x^* \approx \bar{x} \approx 1.6114$. Está terminada a segunda iteração.

3.^a iteração

Resulta da 2.^a iteração que $x_1 = 0.1$, $x_2 = 1.6114$, $x_3 = 1.8691$, $f_1 \approx 108.5625$,

$f_2 \approx 100.0387$ e $f_3 \approx 100.3993$.

Passo 3

$$\bar{x} = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{2[(x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3]},$$

$$\bar{x} = \frac{(1.6114^2 - 1.8691^2) \times 108.5625 + (1.8691^2 - 0.1^2) \times 100.0387 + (0.1^2 - 1.6114^2) \times 100.3993}{2[(1.6114 - 1.8691) \times 108.5625 + (1.8691 - 0.1) \times 100.0387 + (0.1 - 1.6114) \times 100.3993]},$$

$$\bar{x} \approx 1.5644,$$

$$\bar{f} = f(1.5644) = 15(4 - 1.5644) + 25\sqrt{1.5644^2 + 4} \approx 100.0131.$$

Passo 4

Como $x_1 < \bar{x} < x_2$ e $\bar{f} < f_2$, então:

- $x_3 = x_2 \approx 1.6114$,
- $f_3 = f_2 \approx 100.0387$,
- $x_2 = \bar{x} \approx 1.5644$,
- $f_2 = \bar{f} \approx 100.0131$,
- os valores de x_1 e f_1 mantêm-se ($x_1 = 0.1$ e $f_1 \approx 108.5625$).

Uma aproximação para o minimizante $x^* \approx \bar{x} \approx 1.5644$. Esta concluída a terceira iteração.

Deste modo, o preço da colocação da conduta é mínimo para $x \approx 1.5644$.

Observação 3.5 A solução exata deste problema é $x^* = 1.5$. □

Problema 3.2 Admita que às zero horas foi administrado a um doente um fármaco, o Pentacol.

Sabe-se que a concentração do fármaco t horas após ter sido administrado é dada, em miligramas por litro de sangue (mg/l), por

$$c(t) = t^2 \times 1.05^{-5t}.$$

Quando ocorrer a concentração máxima, o doente deve tomar uma segunda dose de Pentacol. Determine a que horas deve ser tomada a segunda dose. Utilize o método de

interpolação quadrática, considerando o intervalo inicial de incerteza $[0, 50]$ e a tolerância de otimização $\varepsilon = 10^{-4}$.

Resolução:

Uma vez que o maximizante da função c é o minimizante da função $-c$, para resolver o problema proposto, vamos determinar o minimizante de $-c$, recorrendo ao programa em MATLAB `interpolacao_quadratica.m`. Para o efeito, no editor do MATLAB, escrevemos o seguinte:

```
interpolacao_quadratica(inline('-x^2*1.05^(- 5*x)', 'x'), 0, 50, 10^(-4))
```

O resultado obtido é o seguinte:

```
PROGRAMA interpolacao_quadratica.m
Valor aproximado para o MINIMIZANTE: 8.198374
Valor aproximado para o MÍNIMO: -9.096335
Número de ITERAÇÕES realizadas: 10
```

Assim sendo, o maximizante da função c é $t \approx 8.198374$. Pelo que a segunda dose de Pentacol deve ser tomada cerca de 8.198374 horas depois da primeira, ou seja, às 8h12min, aproximadamente.

Observação 3.6 Os problemas 3.1 e 3.2 podem ser facilmente reformulados para não ser possível resolvê-los exatamente, com recurso ao Teorema 1.8 (Teorema de Fermat). \square

3.1.2 Interpolação em Dois Pontos

Na Subsecção 3.1.1, estudamos a interpolação quadrática com três pontos. Esse caso implicava, por isso, o conhecimento do valor da função objetivo em três pontos distintos. A abordagem que agora iniciamos diz respeito à interpolação em dois pontos e implica o conhecimento do valor da função objetivo em dois pontos distintos e o da sua derivada em um deles.

Consideremos que a função objetivo f é unimodal no intervalo $[a, b]$ e $f \in \mathcal{C}^1([a, b])$. À semelhança da interpolação em três pontos, o polinómio aproximante é

$$p(x) = a_2x^2 + a_1x + a_0, \quad (3.8)$$

onde a_0 , a_1 e a_2 são constantes, sendo $a_2 \neq 0$. Já vimos que o minimizante de $p(x)$ é

$$\bar{x} = -\frac{a_1}{2a_2}. \quad (3.9)$$

Vamos determinar os valores de a_1 e a_2 , sabendo que $p(x)$ interpola f em x_1 e x_2 ($x_1 \neq x_2$), osculando f em x_1 . Portanto, sendo $f(x_1) = f_1$, $f(x_2) = f_2$ e $f'(x_1) = f'_1$, tem-se

$$\begin{cases} a_2x_1^2 + a_1x_1 + a_0 = f_1 \\ a_2x_2^2 + a_1x_2 + a_0 = f_2 \\ 2a_2x_1 + a_1 = f'_1 \end{cases} \quad (3.10)$$

A solução deste sistema dá-nos os valores de a_0 , a_1 e a_2 . Escrevendo sistema na forma matricial obtém-se

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ 2x_1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f'_1 \end{bmatrix}. \quad (3.11)$$

Para resolver a Equação (3.11), vamos recorrer ao **MATLAB** e considerar, apenas no programa usado, que $f'_1 = g_1$. Para obter os valores de a_0 , a_1 e a_2 usamos o seguinte programa:

```
clear global
syms x1 x2 x3 f1 f2 g1 a0 a1 a2
M=[x1^2 x1 1; x2^2 x2 1; 2*x1 1 0]
F=[f1; f2; g1];
%g1=f'1
A=inv(M)*F;
a2=simplify(A(1))
a1=simplify(A(2))
a0=simplify(A(3))
```

De onde se obtém

```
a2 =
-(f1 - f2 - g1*x1 + g1*x2)/(x1 - x2)^2
a1 =
(2*f1*x1 - 2*f2*x1 - g1*x1^2 + g1*x2^2)/(x1 - x2)^2
a0 =
(g1*x1^2*x2 + f2*x1^2 - g1*x1*x2^2 - 2*f1*x1*x2 + f1*x2^2)/(x1 - x2)^2
```

Como

$$a_1 = \frac{2f_1x_1 - 2f_2x_1 - f'_1x_1^2 + f'_1x_2^2}{(x_1 - x_2)^2} \quad (3.12)$$

e

$$a_2 = -\frac{f_1 - f_2 - f'_1 x_1 + f'_1 x_2}{(x_1 - x_2)^2}, \quad (3.13)$$

da Igualdade (3.9) obtém-se

$$\begin{aligned} \bar{x} &= -\frac{\frac{2f_1 x_1 - 2f_2 x_1 - f'_1 x_1^2 + f'_1 x_2^2}{(x_1 - x_2)^2}}{2\left(-\frac{f_1 - f_2 - f'_1 x_1 + f'_1 x_2}{(x_1 - x_2)^2}\right)}, \\ \bar{x} &= \frac{2f_1 x_1 - 2f_2 x_1 - f'_1 x_1^2 + f'_1 x_2^2}{2(f_1 - f_2 - f'_1 x_1 + f'_1 x_2)}, \\ \bar{x} &= \frac{2x_1(f_1 - f_2) + (x_2^2 - x_1^2)f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}, \\ \bar{x} &= \frac{2x_1(f_1 - f_2) + (x_2^2 - x_1^2 + 2x_1 x_2 - 2x_1 x_2 + 2x_1^2 - 2x_1^2)f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}, \\ \bar{x} &= \frac{2x_1(f_1 - f_2) + (x_2^2 - 2x_1 x_2 + x_1^2)f'_1 + (2x_1 x_2 - 2x_1^2)f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}, \\ \bar{x} &= \frac{2x_1(f_1 - f_2) + (x_2 - x_1)^2 f'_1 + 2x_1(x_2 - x_1)f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}, \\ \bar{x} &= \frac{2x_1[f_1 - f_2 + (x_2 - x_1)f'_1] + (x_2 - x_1)^2 f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}, \\ \bar{x} &= x_1 + \frac{(x_2 - x_1)^2 f'_1}{2[f_1 - f_2 + (x_2 - x_1)f'_1]}. \end{aligned} \quad (3.14)$$

Uma fórmula de interpolação em apenas dois pontos x_1 e x_2 pode ser obtida assumindo $p'(x_1) = f'_1$ e $p'(x_2) = f'_2$. Considerando $f'(x_1) = f'_1$ e $f'_2(x_2) = f'_2$, obtém-se o seguinte sistema de equações:

$$\begin{cases} 2a_2 x_1 + a_1 = f'_1 \\ 2a_2 x_2 + a_1 = f'_2 \end{cases}. \quad (3.15)$$

Observação 3.7 Não é preciso conhecer a_0 para determinar o minimizante de $p(x)$, pelo que não precisamos desse valor. No entanto, a_0 seria preciso se pretendêssemos determinar o mínimo de $p(x)$. \square

Resolvendo o sistema (3.15) pelo método de substituição, obtém-se

$$\begin{cases} a_1 = f'_1 - 2a_2x_1 \\ 2a_2x_2 + a_1 = f'_2 \end{cases},$$

$$\begin{cases} a_1 = f'_1 - 2a_2x_1 \\ 2a_2x_2 + f'_1 - 2a_2x_1 = f'_2 \end{cases},$$

$$\begin{cases} a_1 = f'_1 - 2a_2x_1 \\ a_2 = \frac{f'_2 - f'_1}{2(x_2 - x_1)} \end{cases},$$

$$\begin{cases} a_1 = f'_1 - 2 \times \frac{f'_2 - f'_1}{2(x_2 - x_1)} \times x_1 \\ a_2 = \frac{f'_2 - f'_1}{2(x_2 - x_1)} \end{cases},$$

$$\begin{cases} a_1 = \frac{2(x_2 - x_1)f'_1 - 2x_1(f'_2 - f'_1)}{2(x_2 - x_1)} \\ a_2 = \frac{f'_2 - f'_1}{2(x_2 - x_1)} \end{cases}. \quad (3.16)$$

Combinando as igualdades (3.9) e (3.16), resulta

$$\begin{aligned}
 \bar{x} &= -\frac{\frac{2(x_2 - x_1)f'_1 - 2x_1(f'_2 - f'_1)}{2(x_2 - x_1)}}{2\frac{f'_2 - f'_1}{2(x_2 - x_1)}}, \\
 \bar{x} &= -\frac{(x_2 - x_1)f'_1 - x_1(f'_2 - f'_1)}{f'_2 - f'_1}, \\
 \bar{x} &= -\frac{x_2f'_1 - x_1f'_1 - x_1f'_2 + x_1f'_1}{f'_2 - f'_1}, \\
 \bar{x} &= \frac{x_2f'_1 - x_1f'_2}{f'_1 - f'_2}, \\
 \bar{x} &= \frac{x_2f'_1 - x_2f'_2 + x_2f'_2 - x_1f'_2}{f'_1 - f'_2}, \\
 \bar{x} &= \frac{x_2(f'_1 - f'_2) + (x_2 - x_1)f'_2}{f'_1 - f'_2}, \\
 \bar{x} &= x_2 + \frac{(x_2 - x_1)f'_2}{f'_1 - f'_2}. \tag{3.17}
 \end{aligned}$$

3.2 Método de Interpolação Cúbica

Consideremos que a função objetivo f é unimodal no intervalo $[a, b]$. O método de interpolação cúbica baseia-se na obtenção de um polinómio do terceiro grau

$$p(x) = a_3x^3 + a_2x^2 + a_1x + a_0, \tag{3.18}$$

onde a_0, a_1, a_2 e a_3 são constantes, sendo $a_3 \neq 0$. $p(x)$ é um polinómio aproximante para $f(x)$ em $[a, b]$.

Tal como no método de interpolação quadrática, os coeficientes de $p(x)$, a_0, a_1, a_2 e a_3 , podem ser calculados desde que, em certos pontos, $p(x)$ e/ou $p'(x)$ seja igual a $f(x)$ e/ou $f'(x)$. Uma vez que $p(x)$ tem quatro coeficientes, são necessárias quatro equações para caraterizar completamente $p(x)$. Tais equações podem ser escolhidas de várias maneiras, gerando-se outras tantas fórmulas diferentes de interpolação cúbica.

A representação gráfica de $p(x)$ pode assumir uma das formas representadas na Figura 3.3. Como se pode ver nessa figura, $p(x)$ pode ter máximo ou mínimo locais. Os extremos de $p(x)$, em \mathbb{R} , ocorrem nos pontos x tais que

$$p'(x) = 0, \quad (3.19)$$

de acordo com o Teorema 1.8 (pág. 20). Resolvendo a Equação (3.19) obtém-se

$$3a_3x^2 + 2a_2x + a_1 = 0,$$

$$x = \frac{-2a_2 + \sqrt{(2a_2)^2 - 4 \times 3a_3a_1}}{2 \times 3a_3} \vee x = \frac{-2a_2 - \sqrt{(2a_2)^2 - 4 \times 3a_3a_1}}{2 \times 3a_3},$$

$$x = \frac{-a_2 + \sqrt{a_2^2 - 3a_3a_1}}{3a_3} \vee x = \frac{-a_2 - \sqrt{a_2^2 - 3a_3a_1}}{3a_3}. \quad (3.20)$$

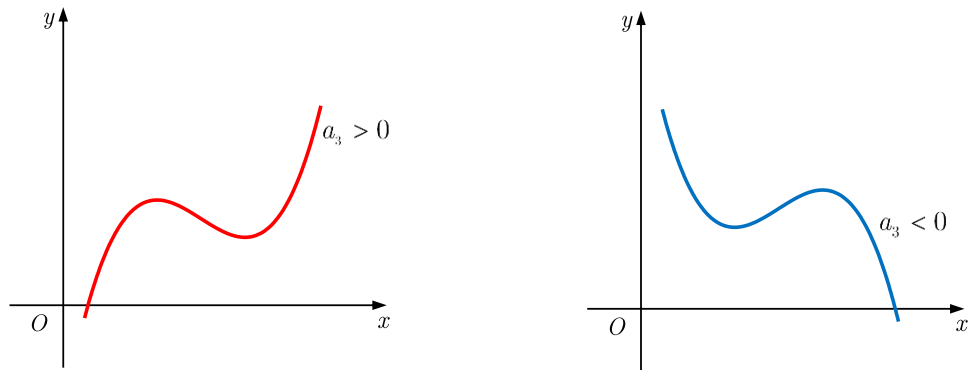


Figura 3.3: Possíveis formas do gráfico do polinómio interpolador de grau 3.

Seja x uma solução da Equação (3.19). Se $p''(x) > 0$, então x é um minimizante de $p(x)$, de acordo com o Teorema 1.9 (pág. 22). Portanto, o minimizante \bar{x} de $p(x)$ é tal que

$$p''(\bar{x}) > 0,$$

$$6a_3\bar{x} + 2a_2 > 0,$$

$$3a_3\bar{x} + a_2 > 0. \quad (3.21)$$

Logo, o minimizante \bar{x} de $p(x)$ é a solução da Equação (3.19) que verifica a Desigualdade (3.21).

Sejam x_1, x_2 e x_3 três pontos distintos de $[a, b]$ e assumamos que f é diferenciável em $[a, b]$. Admitamos ainda que $f(x_1) = f_1, f(x_2) = f_2, f(x_3) = f_3$ e $f'(x_1) = f'_1$.

Uma das muitas possibilidades de obter os coeficientes de $p(x)^3$ é considerar que

$$\begin{cases} a_3x_1^3 + a_2x_1^2 + a_1x_1 + a_0 = f_1 \\ a_3x_2^3 + a_2x_2^2 + a_1x_2 + a_0 = f_2 \\ a_3x_3^3 + a_2x_3^2 + a_1x_3 + a_0 = f_3 \\ 3a_3x_1^2 + 2a_2x_1 + a_1 = f'_1. \end{cases} \quad (3.22)$$

Usando matrizes, obtém-se a seguinte equação:

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ 3x_1^2 & 2x_1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f'_1 \end{bmatrix} \quad (3.23)$$

A resolução da Equação (3.23) pode ser feita recorrendo ao **MATLAB**. O programa seguinte permite obter os valores de a_0, a_1, a_2 e a_3 (apenas no programa, consideramos que $f'_1 = g_1$).

```
clear global
syms x1 x2 x3 f1 f2 f3 g1 a0 a1 a2 a3
M=[x1^3 x1^2 x1 1; x2^3 x2^2 x2 1; x3^3 x3^2 x3 1; 3*x1^2 2*x1 1 0]
F=[f1; f2; f3; g1];
%Consideramos que g1=f'1
A=inv(M)*F;
a3=simplify(A(1))
a2=simplify(A(2))
a1=simplify(A(3))
a0=simplify(A(4))
```

Por serem expressões demasiado longas, optamos por omitir o resultado do programa

³Alternativamente, por exemplo, poderíamos considerar que $p(x)$ interpola f em quatro pontos distintos. Resultando, daí, um sistema de Vandermonde.

anterior. Considerando:

$$\beta = \frac{f_2 - f_1 + (x_1 - x_2)f'_1}{(x_1 - x_2)^2}, \quad (3.24)$$

$$\gamma = \frac{f_3 - f_1 + (x_1 - x_3)f'_1}{(x_1 - x_3)^2}, \quad (3.25)$$

$$\theta = \frac{2x_1^2 - x_2(x_1 + x_2)}{x_1 - x_2}, \quad (3.26)$$

$$\psi = \frac{2x_1^2 - x_3(x_1 + x_3)}{x_1 - x_3}, \quad (3.27)$$

o resultado do programa, para a_1 , a_2 e a_3 , pode ser reescrito do seguinte modo:

$$a_3 = \frac{\beta - \gamma}{\theta - \psi}, \quad (3.28)$$

$$a_2 = \beta - \theta a_3, \quad (3.29)$$

$$a_1 = f'_1 - 2a_2x_1 - 3a_3x_1^2. \quad (3.30)$$

Conhecidos os coeficientes do polinómio $p(x)$, podemos, então, determinar \bar{x} , considerando, como já foi referido atrás, as condições (3.19) (ou (3.20)) e (3.21).

O algoritmo 3.2 permite operacionalizar método de interpolação cúbica.

Neste algoritmo, no *passo 5*, o intervalo que contém o minimizante x^* de f é mantido, substituindo o ponto (x_1 ou x_2 ou x_3) que gera a maior imagem por f . Esse ponto é substituído pela estimativa \bar{x} do minimizante. No caso do ponto a ser substituído ser x_1 , a derivada $f'(x_1)$ deve ser recalculada uma vez que é necessária para o cálculo de a_1 , β e γ .

As igualdades de (3.24) a (3.30) mostram que uma iteração do método de interpolação cúbica implica muitos mais cálculos do que uma iteração do método de interpolação quadrática. No entanto, o primeiro pode ser mais eficiente, dado que a aproximação de f por um polinómio de grau 3 é, em princípio, mais precisa do que por um polinómio de grau 2. Como consequência, a convergência é alcançada num número pequeno de iterações e o método é mais tolerante a perdas eventuais do intervalo de incerteza.

Apresenta-se a seguir uma rotina **MATLAB** que implementa o algoritmo 3.2. Esta ferramenta será útil na resolução de problemas de minimização usando o método de interpolação cúbica e foi adaptada da função *cubic_inter* proposta por [1, Antoniou e Lou], passando a designar-se por *interpolacao_cubica*.

Algoritmo 3.2: Método de Interpolação Cúbica

Passo 1

Entrada: x_1, x_2, x_3 , tolerância ε e função objetivo f .

Passo 2

Considerar $\bar{x}_0 = 10^{308}$. Calcular:

$$f_1 = f(x_1); f_2 = f(x_2); f_3 = f(x_3); f'_1 = f'(x_1).$$

Passo 3

Calcular as constantes β, γ, θ e ψ , usando as igualdades de (3.24) a (3.27).

Calcular as constantes a_1, a_2 e a_3 , usando as igualdades de (3.28) a (3.30).

Calcular os extremantes de $p(x)$ usando a condição (3.20).

Selecionar o minimizante de $p(x)$ usando a Desigualdade (3.21).

Calcular $\bar{f} = f(\bar{x})$.

Passo 4

se $|\bar{x} - \bar{x}_0| < \varepsilon$ **então**

$x^* = \bar{x}$ e $f(x^*) = \bar{f}$.

 Stop.

Passo 5

Determinar m tal que $f_m = \max\{f_1, f_2, f_3\}$.

Considerar:

$$\bar{x}_0 = \bar{x}.$$

$$x_m = \bar{x}.$$

$$f_m = \bar{f}.$$

se $m = 1$ **então**

$f'_1 = f'(\bar{x})$

Voltar ao **Passo 3**.

```
% Programa: interpolacao_cubica.m
% Título: Método de Interpolação Cúbica
% Descrição: Implementação do Método de Interpolação Cúbica
% Dados de entrada:
%   fname: função objetivo
%   gname: derivada da função objetivo
%   [x1,x3]: intervalo inicial de incerteza
%   epsi: tolerância de otimização
% Dados de saída:
%   xs: minimizante
%   fs: mínimo da função objetivo
%   k: número de iterações necessárias
%   kg: número de avaliações da primeira derivada
% Exemplo:
```

```

% Determinar o mínimo de  $f(x)=x\sin(x)$ ,
% sabendo que  $4 \leq x \leq 5$ .
% Solução:
% Executar o comando
% interpolacao_cubica(inline('x*sin(x)','x'),inline
% ('x*cos(x)+sin(x)','x'),4,5,0.03)
%-----
function interpolacao_cubica(fname,gname,x1,x3,epsi)
disp(' ')
disp('PROGRAMA interpolacao_cubica.m')
x0bar = 1e308;
% Realização da 1.a iteração
x2 = 0.5*(x1+x3);
f1 = feval(fname,x1);
f2 = feval(fname,x2);
f3 = feval(fname,x3);
g1 = feval(gname,x1);
kg = 1;
dx12 = x1 - x2;
dx13 = x1 - x3;
sx12 = x1 + x2;
sx13 = x1 + x3;
bt = (f2 - f1 + g1*dx12)/(dx12^2);
gm = (f3 - f1 + g1*dx13)/(dx13^2);
th = (2*x1^2 - x2*sx12)/dx12;
ps = (2*x1^2 - x3*sx13)/dx13;
a3 = (bt - gm)/(th - ps);
a2 = bt - th*a3;
a1 = g1 - 2*a2*x1 - 3*a3*x1^2;
sw = sqrt(a2^2 - 3*a1*a3);
xe1 = (-a2+sw)/(3*a3);
xe2 = (-a2-sw)/(3*a3);
ct = -a2/(3*a3);
if a3 > 0,
    if xe1 > ct,
        xbar = xe1;
    else
        xbar = xe2;
    end
else
    if xe1 < ct,
        xbar = xe1;
    else

```

```
        xbar = xe2;
    end
end
fbar = feval(fname,xbar);
d = abs(x0bar-xbar);
k = 1;
% Realização da busca baseada na interpolação cúbica
while d >= epsi,
    if x1 < xbar & xbar < x2,
        if fbar <= f2,
            x3 = x2;
            f3 = f2;
            x2 = xbar;
            f2 = fbar;
        else
            x1 = xbar;
            f1 = fbar;
            g1 = feval(gname,x1);
            kg = kg + 1;
        end
    elseif x2 < xbar & xbar < x3,
        if fbar <= f2,
            x1 = x2;
            f1 = f2;
            g1 = feval(gname,x1);
            kg = kg + 1;
            x2 = xbar;
            f2 = fbar;
        else
            x3 = xbar;
            f3 = fbar;
        end
    end
    x0bar = xbar;
    dx12 = x1 - x2;
    dx13 = x1 - x3;
    sx12 = x1 + x2;
    sx13 = x1 + x3;
    bt = (f2 - f1 + g1*dx12)/(dx12^2);
    gm = (f3 - f1 + g1*dx13)/(dx13^2);
    th = (2*x1^2 - x2*sx12)/dx12;
    ps = (2*x1^2 - x3*sx13)/dx13;
    a3 = (bt - gm)/(th - ps);
```

```

a2 = bt - th*a3;
a1 = g1 - 2*a2*x1 - 3*a3*x1^2;
sw = sqrt(a2^2 - 3*a1*a3);
xe1 = (-a2+sw)/(3*a3);
xe2 = (-a2-sw)/(3*a3);
ct = -a2/(3*a3);
if a3 > 0,
    if xe1 > ct,
        xbar = xe1;
    else
        xbar = xe2;
    end
else
    if xe1 < ct,
        xbar = xe1;
    else
        xbar = xe2;
    end
end
fbar = feval(fname,xbar);
d = abs(x0bar-xbar);
k = k + 1;
end
% Apresentação de resultados
xs = real(xbar);%
fs = real(fbar);
fprintf('Valor aproximado para o MINIMIZANTE: ');
fprintf(' %.6f\n', xs);
fprintf('Valor aproximado para o MÍNIMO: %.6f\n',fs);
fprintf('Número de ITERAÇÕES realizadas: %d\n',k);
fprintf('Número de AVALIAÇÕES da 1.ª derivada da função objetivo:
%d',kg);
end

```

Para finalizar a análise do método de interpolação cúbica, vamos aplicar o algoritmo 3.2 à resolução de problemas.

Problema 3.3 *Duas povoações, A e B, distanciadas de 8 Km uma da outra, estão a igual distância de uma fonte de abastecimento de água, localizada em F.*

Pretende-se construir uma canalização ligando a fonte às duas povoações, como se indica na Figura 3.4.

A canalização é formada por três canos: um que vai da fonte F até um ponto P e dois que partem de P , um para A e outro para B . O ponto P está a igual distância de A e de B .

Tem-se ainda que:

- o ponto M , ponto médio de $[AB]$, dista 4 Km de F ;
- x é a amplitude do ângulo PAM ($x \in [0, \frac{\pi}{4}]$).

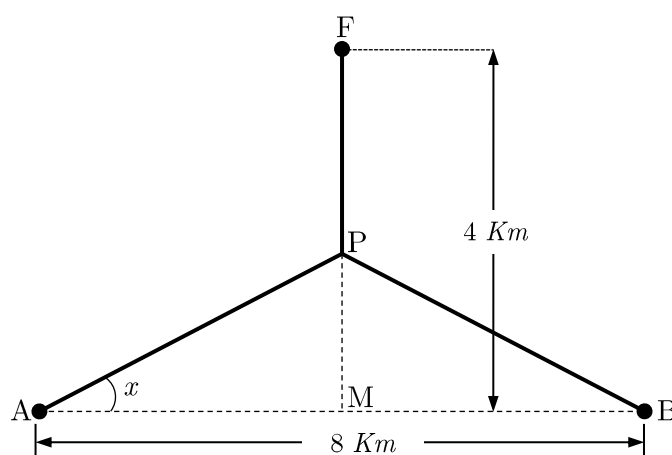


Figura 3.4: Esquematização do Problema 3.3.

Determine o valor de x , em radianos, arredondado às décimas, para o qual o comprimento total da canalização é mínimo. Indique, nesse caso, o comprimento da canalização, em quilômetros, arredondado às décimas. Utilize o método de interpolação cúbica, considerando a tolerância de otimização $\varepsilon = 10^{-4}$.

Resolução:

Uma vez que não dispomos da expressão da função objetivo, vamos começar por determiná-la. Seja f essa função.

Tem-se:

$$\cos x = \frac{\overline{MA}}{\overline{PA}} = \frac{4}{\overline{PA}} \text{ pelo que } \overline{PA} = \frac{4}{\cos x}$$

$$\tan x = \frac{\overline{PM}}{\overline{MA}} = \frac{\overline{PM}}{4} \text{ pelo que } \overline{PM} = 4 \tan x \text{ donde resulta } \overline{FP} = 4 - 4 \tan x$$

O comprimento da canalização é igual a $\overline{FP} + 2 \times \overline{PA}$. Uma vez que $f(x)$ representa o comprimento da canalização em função do ângulo de amplitude x , tem-se que

$$f(x) = 4 - 4 \tan x + \frac{8}{\cos x}.$$

Para resolver este problema, vamos recorrer à função `interpolacao_cubica`, considerando que a tolerância de otimização é 10^{-4} e que $f'(x) = \frac{8 \sin(x) - 4}{\cos^2 x}$. No editor do MATLAB, escrevemos

```
interpolacao_cubica(inline('4-4*tan(x)+ 8/cos (x)', 'x'),
inline('(8*sin(x) - 4)/(cos(x))^2', 'x'), 0, pi/4, 10^-4)
```

O resultado obtido é o seguinte:

```
PROGRAMA interpolacao_cubica.m
Valor aproximado para o MINIMIZANTE: 0.523598
Valor aproximado para o MÍNIMO: 10.928203
Número de ITERAÇÕES realizadas: 4
Número de AVALIAÇÕES da 1.ª derivada da função objetivo: 4
```

O comprimento mínimo da canalização é 10.9 Km e obtém-se para $x \approx 0.5$ radianos.

Observação 3.8 Apesar de a precisão do resultado parecer grosseira, deve ser suficiente em engenharia. Até porque a função varia pouco no ponto de estacionariedade. \square

Problema 3.4 Em engenharia do ambiente, o nível c de concentração de oxigénio, em miligramas por litro, existente num rio, a x Km de um local de descarga de poluentes, é habitualmente calculado pela expressão

$$c(x) = 10 - 20(e^{-0.2x} - e^{-0.75x}).$$

Determine a distância do ponto de descarga de poluentes onde a concentração de oxigénio é mínima. Qual é o valor da concentração de oxigénio nesse ponto? Realize duas iterações do método de interpolação cúbica, considerando o intervalo inicial de incerteza $[0, 4]$.

Resolução:

Para efeitos desta resolução, assume-se que função $c = f$.

1.^a iteração**Passo 1**

Como o intervalo de incerteza inicial é $[0, 4]$, $x_1 = 0$ e $x_3 = 4$.

$$x_2 = \frac{x_1 + x_3}{2} = \frac{0 + 4}{2} = 2.$$

Passo 2

$$f_1 = f(0) = 10 - 20(e^{-0.2 \times 0} - e^{-0.75 \times 0}) = 10,$$

$$f_2 = f(2) = 10 - 20(e^{-0.2 \times 2} - e^{-0.75 \times 2}) \approx 1.0562,$$

$$f_3 = f(4) = 10 - 20(e^{-0.2 \times 4} - e^{-0.75 \times 4}) \approx 2.0092,$$

$$f'(x) = 4e^{-0.2x} - 15e^{-0.75x}.$$

$$f'_1 = f'(0) = 4e^{-0.2 \times 0} - 15e^{-0.75 \times 0} = -11$$

Passo 3

$$\beta = \frac{f_2 - f_1 + (x_1 - x_2)f'_1}{(x_1 - x_2)^2} \approx \frac{1.0562 - 10 + (0 - 2) \times (-11)}{(0 - 2)^2} \approx 3.2641,$$

$$\gamma = \frac{f_3 - f_1 + (x_1 - x_3)f'_1}{(x_1 - x_3)^2} \approx \frac{2.0092 - 10 + (0 - 4) \times (-11)}{(0 - 4)^2} \approx 2.2506,$$

$$\theta = \frac{2x_1^2 - x_2(x_1 + x_2)}{x_1 - x_2} \approx \frac{2 \times 0^2 - 2 \times (0 + 2)}{0 - 2} = 2,$$

$$\psi = \frac{2x_1^2 - x_3(x_1 + x_3)}{x_1 - x_3} = \frac{2 \times 0^2 - 4 \times (0 + 4)}{0 - 4} = 4,$$

$$a_3 = \frac{\beta - \gamma}{\theta - \psi} \approx \frac{3.2641 - 2.2506}{2 - 4} \approx -0.5067,$$

$$a_2 = \beta - \theta a_3 \approx 3.2641 - 2 \times (-0.5067) = 4.2775,$$

$$a_1 = f'_1 - 2a_2x_1 - 3a_3x_1^2 \approx -11 - 2 \times 4.2775 \times 0 - 3 \times (-0.5067) \times 0^2 = -11.$$

Cálculo dos extremantes:

$$x = \frac{-a_2 + \sqrt{a_2^2 - 3a_3a_1}}{3a_3} \vee x = \frac{-a_2 - \sqrt{a_2^2 - 3a_3a_1}}{3a_3},$$

$$x = \frac{-a_2 + \sqrt{a_2^2 - 3a_3a_1}}{3a_3} = \frac{4.2775 + \sqrt{(4.2775)^2 - 16.7211}}{3 \times (-0.5067)} \approx 1.9883, \text{ ou}$$

$$x = \frac{-a_2 - \sqrt{a_2^2 - 3a_3a_1}}{3a_3} = \frac{4.2775 - \sqrt{(4.2775)^2 - 16.7211}}{3 \times (-0.5067)} \approx 3.6393.$$

Escolha do minimizante:

$$3a_3\bar{x} + a_2 > 0,$$

$$3 \times (-0.5067) \times 3.6393 + 4.2775 \approx -1.2546 < 0 \quad 3 \times (-0.5067) \times 1.9883 + 4.2775 \approx 1.2551 > 0,$$

Então, $\bar{x} \approx 1.9883$.

Passo 4

Não se aplica, porque vamos realizar duas iterações.

Passo 5

$$f_1 = 10,$$

$$f_2 \approx 1.0562,$$

$$f_3 \approx 2.0092, \max\{f_1, f_2, f_3\} = f_1. \text{ Então, } m = 1 \text{ e:}$$

- $\bar{x}_0 = \bar{x} \approx 1.9883,$
- $x_m = x_1 = \bar{x} \approx 1.9883,$
- $f_m = f_1 = \bar{f} = f(1.2174) = 10 - 20(e^{-0.2 \times 1.9883} - e^{-0.75 \times 1.9883}) \approx 1.0641,$
- $f'_1 = f'(\bar{x}) = f'(1.9883) = 4e^{-0.2 \times 1.9883} - 15e^{-0.75 \times 1.9883} \approx -0.6889.$

2.ª iteração

Passo 1 e Passo 2

Da iteração anterior resulta que:

- $x_1 \approx 1.9883$,
- $x_2 = 2$,
- $x_3 = 4$,
- $f_1 \approx 1.0641$,
- $f_2 \approx 1.0562$,
- $f_3 \approx 2.0092$,
- $f'_1 \approx -0.6889$.

Passo 3

$$\beta = \frac{f_2 - f_1 + (x_1 - x_2)f'_1}{(x_1 - x_2)^2} \approx \frac{1.0562 - 1.0641 + (1.9883 - 2) \times (-0.6889)}{(1.9883 - 2)^2} \approx 1.1698,$$

$$\gamma = \frac{f_3 - f_1 + (x_1 - x_3)f'_1}{(x_1 - x_3)^2} \approx \frac{2.0092 - 1.0641 + (1.9883 - 4) \times (-0.6889)}{(1.9883 - 4)^2} \approx 0.5760,$$

$$\theta = \frac{2x_1^2 - x_2(x_1 + x_2)}{x_1 - x_2} \approx \frac{2 \times 1.9883^2 - 2 \times (1.9883 + 2)}{1.9883 - 2} \approx 5.9766,$$

$$\psi = \frac{2x_1^2 - x_3(x_1 + x_3)}{x_1 - x_3} \approx \frac{2 \times 1.9883^2 - 4 \times (1.9883 + 4)}{1.9883 - 4} \approx 7.9766,$$

$$a_3 = \frac{\beta - \gamma}{\theta - \psi} \approx \frac{1.1698 - 0.5760}{5.9766 - 7.9766} \approx -0.2969,$$

$$a_2 = \beta - \theta a_3 \approx 1.1698 - 5.9766 \times (-0.2969) \approx 2.9442,$$

$$a_1 = f'_1 - 2a_2x_1 - 3a_3x_1^2,$$

$$a_1 \approx -0.6889 - 2 \times 2.9442 \times 1.9883 - 3 \times (-0.2969) \times 1.9883^2 \approx -8.8756,$$

Cálculo dos extremantes:

$$x = \frac{-a_2 + \sqrt{a_2^2 - 3a_3a_1}}{3a_3} \vee x = \frac{-a_2 - \sqrt{a_2^2 - 3a_3a_1}}{3a_3},$$

$$x = \frac{-a_2 + \sqrt{a_2^2 - 3a_3a_1}}{3a_3} \approx \frac{-2.9442 + \sqrt{(-2.9442)^2 - 7.90549692}}{3 \times (-0.2969)} \approx 2.3249, \text{ ou}$$

$$x = \frac{-a_2 - \sqrt{a_2^2 - 3a_3a_1}}{3a_3} \approx \frac{-2.9442 - \sqrt{(-2.9442)^2 - 7.90549692}}{3 \times (-0.2969)} \approx 4.2862,$$

Escolha do minimizante:

$$3a_3\bar{x} + a_2 > 0,$$

$$3 \times (-0.2969) \times 2.3249 + 2.9442 \approx 0.8734 > 0,$$

$$3 \times (-0.2969) \times 4.2862 + 2.9442 \approx -0.8734 < 0,$$

Então, $\bar{x} \approx 2.3249$.

Passo 4

Cumprida a segunda iteração, terminamos a aplicação do algoritmo. Um valor aproximado para o minimizante é, portanto, $x^* = \bar{x} \approx 2.3249$ e para o mínimo é $f(2.3249) \approx 0.9346$.

A concentração de oxigénio é mínima num ponto situado a cerca de 2.3 Km do local de descarga de poluentes, sendo igual a 0.93 mg/l, aproximadamente.

Observação 3.9 A solução exata é $x^* = \frac{20 \ln \left(\frac{20}{21} \right)}{11} = 2.4031924\dots$ e $c(x^*) = 0.9303029\dots$ □

Até aqui precisávamos de conhecer à partida um intervalo $[a, b]$ que continha o minimizante. O algoritmo seguinte encontra o intervalo $[a, b]$ que contém o minimizante mais próximo de zero.

3.3 Método de Davies, Swann e Campey

Este método serve para resolver o problema

$$\min_{x \in \mathbb{R}} f(x) \tag{3.31}$$

onde f é unimodal em \mathbb{R} , isto é, se f é unimodal em qualquer intervalo $[a, b] \subset \mathbb{R}$.

O método devido a Davies, Swann e Campey combina um método de busca com uma aproximação polinomial. O método de busca é usado para estabelecer e manter um intervalo para o minimizante x^* de f , enquanto que o método de aproximação é usado para gerar uma estimativa para x^* .

Neste método, f é avaliada num conjunto de valores crescentes ou decrescentes de x até que se localize x^* num intervalo. Em seguida, a fórmula de interpolação quadrática para pontos igualmente espaçados é usada para aproximar x^* . Este procedimento é repetido várias vezes até se alcançar a precisão desejada.

O algoritmo requer os seguintes dados de entrada:

- um ponto inicial $x_{0,1}$;
- um incremento inicial δ_1 ;
- uma constante de escala C ;
- uma tolerância de otimização ε .

Na iteração k , tem-se um ponto inicial $x_{0,k}$ e um incremento inicial δ_k . Para a iteração seguinte, são necessários um novo ponto inicial $x_{0,k+1}$ e um novo incremento δ_{k+1} .

Inicialmente, f é avaliada nos pontos $x_{0,k} - \delta_k$, $x_{0,k}$ e $x_{0,k} + \delta_k$, de onde pode resultar um dos três casos seguintes, uma vez que f é unimodal em \mathbb{R} :

- (a) $f(x_{0,k} + \delta_k) < f(x_{0,k}) < f(x_{0,k} - \delta_k)$;
- (b) $f(x_{0,k} - \delta_k) < f(x_{0,k}) < f(x_{0,k} + \delta_k)$;
- (c) $f(x_{0,k}) \leq f(x_{0,k} - \delta_k)$ e $f(x_{0,k}) \leq f(x_{0,k} + \delta_k)$.

No caso (a), o minimizante de f localiza-se no sentido positivo e $f(x)$ deve ser avaliada para valores crescentes x até se obter um valor para $f(x)$ que é maior do que o anterior, ao fim de n avaliações de f . Nestas circunstâncias, o intervalo $[x_{0,k}, x_{n,k}]$ é um intervalo de incerteza para x^* . A amplitude do intervalo entre pontos sucessivos é incrementada duplicando a amplitude do intervalo anterior, obtendo-se a sucessão de pontos

$$\begin{aligned}
 & x_{0,k} \\
 x_{1,k} &= x_{0,k} + \delta_k \\
 x_{2,k} &= x_{1,k} + 2\delta_k \\
 x_{3,k} &= x_{2,k} + 4\delta_k \\
 x_{4,k} &= x_{3,k} + 8\delta_k \\
 & \vdots \\
 x_{n,k} &= x_{n-1,k} + 2^{n-1}\delta_k,
 \end{aligned} \tag{3.32}$$

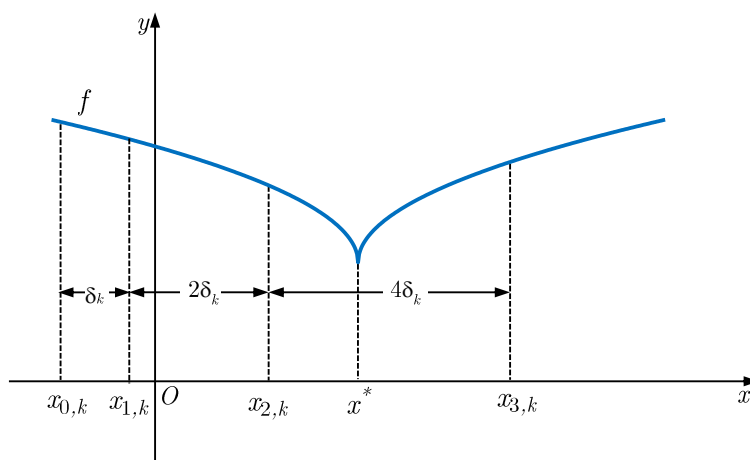


Figura 3.5: Método de busca usado no algoritmo de Davies, Swann e Campey.

como se ilustra na Figura 3.5. Uma vez que a amplitude de cada intervalo é o dobro da amplitude do intervalo anterior, o ponto médio do último intervalo — o intervalo $[x_{n-1,k}, x_{n,k}]$ — é

$$x_m = x_{n-1,k} + \frac{2^{n-1}\delta_k}{2},$$

ou seja,

$$x_m = x_{n-1,k} + 2^{n-2}\delta_k. \quad (3.33)$$

Portanto, os pontos $x_{n-2,k}$, $x_{n-1,k}$, $x_{m,k}$ e $x_{n,k}$ estão igualmente espaçados, pois os intervalos $[x_{n-2,k}, x_{n-1,k}]$, $[x_{n-1,k}, x_{m,k}]$ e $[x_{m,k}, x_{n,k}]$ têm amplitude $2^{n-2}\delta_k$. Acresce que daqueles pontos resulta o intervalo de incerteza $[x_{n-2,k}, x_{n,k}]$, como sugere a Figura 3.6. Consideremos

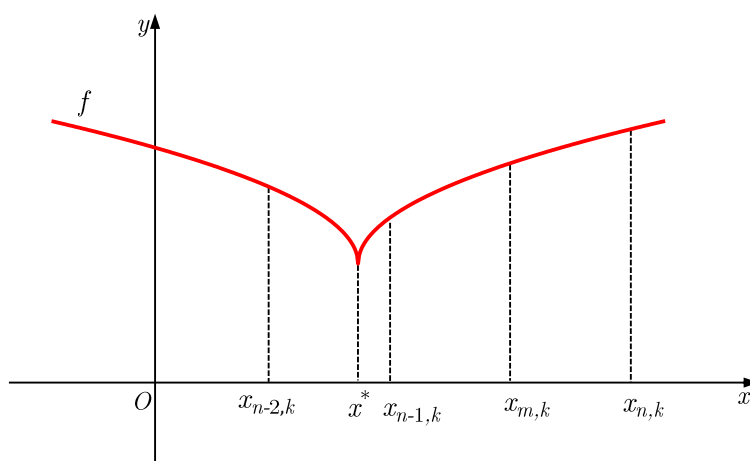


Figura 3.6: Redução do intervalo de incerteza no algoritmo de Davies, Swann e Campey. Caso $f_{m,k} \geq f_{n-1}$.

$f_{n-2,k} = f(x_{n-2,k})$, $f_{n-1,k} = f(x_{n-1,k})$, $f_{m,k} = f(x_{m,k})$ e $f_{n,k} = f(x_{n,k})$.

Se $f_{m,k} \geq f_{n-1,k}$, então o minimizante x^* de f está no intervalo $[x_{n-2,k}, x_{m,k}]$, como se mostra na Figura 3.6. Uma estimativa para x^* , usando a Fórmula (3.7), de interpolação quadrática em três pontos igualmente espaçados, fazendo $x_1 = x_{n-2,k}$, $x_2 = x_{n-1,k}$ e $x_3 = x_{m,k}$, é

$$x_{0,k+1} = x_{n-1,k} + \frac{2^{n-2}\delta_k(f_{n-2,k} - f_{m,k})}{2(f_{n-2,k} - 2f_{n-1,k} + f_{m,k})}. \quad (3.34)$$

Se $f_{m,k} < f_{n-1,k}$, então o minimizante x^* de f está no intervalo $[x_{n-1,k}, x_{n,k}]$, como sugere a Figura 3.7. Uma estimativa para x^* , usando a Fórmula (3.7), de interpolação quadrática em três pontos igualmente espaçados, fazendo $x_1 = x_{n-1,k}$, $x_2 = x_{m,k}$ e $x_3 = x_{n,k}$, é

$$x_{0,k+1} = x_{m,k} + \frac{2^{n-2}\delta_k(f_{n-2,k} - f_{n,k})}{2(f_{n-1,k} - 2f_{m,k} + f_{n,k})}. \quad (3.35)$$

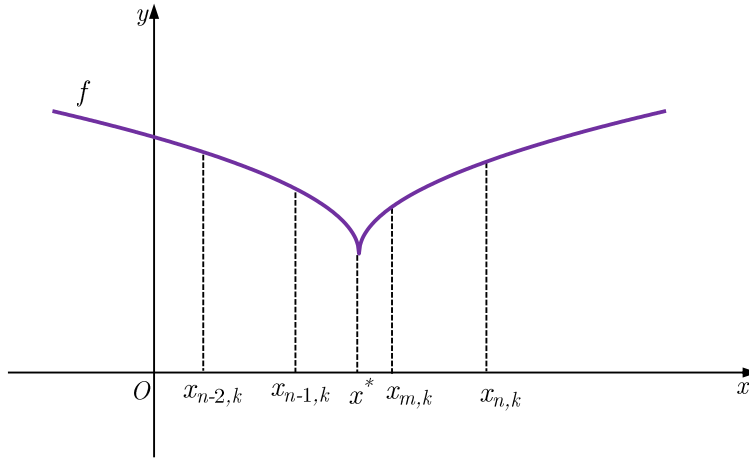


Figura 3.7: Redução do intervalo de incerteza no algoritmo de Davies, Swann e Campey. Caso $f_{m,k} < f_{n-1,k}$.

No caso (b), x^* está localizado no sentido negativo. Portanto, f deve ser avaliada para valores decrescentes de x com passos $\delta_k, 2\delta_k, 4\delta_k, 8\delta_k, \dots$ ($\delta_k < 0$) até que o mínimo de f seja localizado. O procedimento a adotar é similar ao do caso (a), diferindo apenas no facto de no caso (b) se ter $\delta_k < 0$.

Finalmente, no caso (c), x^* está contido no intervalo $[x_{0,k} - \delta_k, x_{0,k} + \delta_k]$. Se considerarmos $f_{-1,k} = f(x_{0,k} - \delta_k)$, $f_{0,k} = f(x_{0,k})$ e $f_{1,k} = f(x_{0,k} + \delta_k)$, fazendo $x_1 = x_{0,k} - \delta_k$, $x_2 = x_{0,k}$ e $x_3 = x_{0,k} + \delta_k$ na Fórmula (3.7), de interpolação quadrática para pontos igualmente espaçados, obtém-se uma estimativa para x^* . Ou seja,

$$x_{0,k+1} = x_{0,k} + \frac{\delta_k(f_{-1,k} - f_{1,k})}{2(f_{-1,k} - 2f_{0,k} + f_{1,k})}. \quad (3.36)$$

Para concluirmos a iteração k temos que definir o novo incremento δ_{k+1} . Para tal, consideramos $\delta_{k+1} = C\delta_k$, com $0 < C < 1$. Podemos, por exemplo, escolher $C = 0.1$. O propósito é ajustar a escala de busca para o novo intervalo. Os princípios que acabamos de apresentar permitem construir o algoritmo 3.3.

O parâmetro δ_1 é uma constante positiva pequena que pode ser igual a $0.1x_{0,1}$. A variável p , que assume os valores 1 ou -1 , usada nos passos 3 e 6, permite que as fórmulas (3.34) e (3.35) possam ser usadas quer quando os valores de x crescem ou decrescem. A constante ε referida no passo 1 determina a precisão da solução. No caso de ε ser muito pequeno, por exemplo, menor que 10^{-6} , como a solução é aproximada, tem-se $f_{n-2,k} \approx f_{n-1,k} \approx f_{n,k} \approx f_{n,k}$. Consequentemente, é evidente a possibilidade de dividir por zero ao calcular $x_{0,k+1}$. No entanto, este problema pode ser prevenido usando verificações adequadas nos passos 6 e 7.

Uma alternativa ao algoritmo 3.3 pode ser obtida substituindo a Fórmula (3.7), de interpolação quadrática para pontos igualmente espaçados, pela Fórmula (3.6). Se isso for feito, a avaliação da função referida no passo 5 é desnecessária.

Outra modificação possível passa pela utilização da interpolação cúbica estudada na Secção 3.2, em vez de interpolação quadrática. Nesse caso, o Algoritmo é suscetível de reduzir o número de avaliações da função objetivo. No entanto, a quantidade de cálculos poderia aumentar devido à formulação mais complexa da interpolação cúbica.

Na resolução de problemas, vamos usar uma rotina MATLAB que implementa o Algoritmo 3.3. Esta ferramenta foi adaptada da função *DSC_search* proposta por [1, Antoniou e Lou], passando a designar-se por *davies_swann_campey*.

```
% Programa: davies_swann_campey.m
% Título: Método de Davies, Swann e Campey
% Descrição: Implementação do método de Davies, Swann e Campey
% Dados de entrada:
%   fname: função objetivo
%   x01: ponto inicial
%   dt1: incremento inicial
%   C: constante de escala (0<C<1)
%   epsi: tolerância de otimização
% Dados de saída:
%   xs: minimizante da função objetivo
%   fs: mínimo da função objetivo
%   k: número de iterações realizadas
%   ke: número de avaliações da função objetivo
% Exemplo:
% Determinar o mínimo de f(x)=x*sin(x),
% sabendo que -2<= x <= 2.
% Solução:
```

Algoritmo 3.3: Método de Davies, Swann e Campey

Passo 1

Entrada: $x_{0,1}, \delta_1$ e C . Tolerância ε . Função objetivo f . Considerar $k = 0$.

Passo 2

Considerar $k = k + 1$; $x_{-1,k} = x_{0,k} - \delta_k$ e $x_{1,k} = x_{0,k} + \delta_k$.
Calcular $f_{0,k} = f(x_{0,k})$ e $f_{1,k} = f(x_{1,k})$.

Passo 3

se $f_{0,k} > f_{1,k}$ **então**
 | Considerar $p = 1$ e ir para o Passo 4.
senão
 | Calcular $f_{-1,k} = f(x_{-1,k})$.
 | **se** $f_{-1,k} < f_{0,k}$ **então**
 | | Considerar $p = -1$ e ir para o Passo 4.
 | **senão se** $f_{-1,k} \geq f_{0,k}$ e $f_{0,k} \leq f_{1,k}$ **então**
 | | Ir para o Passo 7.
fim

fim

Passo 4

Considerar $n = 1$.

repita

| $f_{n,k} = f(x_{n-1,k} + 2^{n-1}p\delta_k)$ e $n = n + 1$.

até $f_{n,k} > f_{n-1,k}$;

Passo 5

Calcular $f_{m,k} = f(x_{n-1,k} + 2^{n-2}p\delta_k)$.

Passo 6

se $f_{m,k} \geq f_{n-1,k}$ **então**

| Calcular $x_{0,k+1} = x_{n-1,k} + \frac{2^{n-2}\delta_k p(f_{n-2,k} - f_{m,k})}{2(f_{n-2,k} - 2f_{n-1,k} + f_{m,k})}$ (Igualdade (3.34))

senão se $f_{m,k} < f_{n-1,k}$ **então**

| Calcular $x_{0,k+1} = x_{m,k} + \frac{2^{n-2}\delta_k p(f_{n-2,k} - f_{n,k})}{2(f_{n-1,k} - 2f_{m,k} + f_{n,k})}$ (Igualdade (3.35))

fim

se $2^{n-2}\delta_k \leq \varepsilon$ **então**

| Ir para o Passo 8.

senão

| Considerar $\delta_{k+1} = C\delta_k$ e ir para o Passo 2.

fim

Passo 7

Calcular $x_{0,k+1} = x_{0,k} + \frac{\delta_k(f_{-1,k} - f_{1,k})}{2(f_{-1,k} - 2f_{0,k} + f_{1,k})}$ (Igualdade (3.36))

se $\delta_k \leq \varepsilon$ **então**

| Ir para o Passo 8.

senão

| $\delta_{k+1} = C\delta_k$ e ir para o Passo 2.

fim

Passo 8

Resultado: $x^* = x_{0,k+1}$ e $f(x^*) = f(x_{0,k+1})$. Stop.

```
% Executar o comando
% davies_swann_campey(inline('x*sin(x)','x'),-2,0.05,0.1,1e-6)
%-----
function davies_swann_campey(fname,x01,dt1,C,epsi)
disp(' ')
disp('PROGRAMA davies_swann_campey.m')
k = 0;
ke = 0;
dt = dt1;
x0 = x01;
er = dt;
while er >= epsi,
    x1 = x0 + dt;
    x_1 = x0 - dt;
    f0 = feval(fname,x0);
    f1 = feval(fname,x1);
    ke = ke + 2;
    if f0 > f1,
        p = 1;
        n = 1;
        fn_1 = f0;
        fn = f1;
        xn = x1;
        ind = 0;
    else
        f_1 = feval(fname,x_1);
        ke = ke + 1;
        if f_1 < f0,
            p = -1;
            n = 1;
            fn_1 = f0;
            fn = f_1;
            xn = x_1;
            ind = 0;
        else
            ind = 1;
        end
    end
end
if ind == 0,
    while fn <= fn_1,
        n = n + 1;
        fn_2 = fn_1;
        fn_1 = fn;
```

```
        xn_1 = xn;
        xn = xn_1 + (2^(n-1))*p*dt;
        fn = feval(fname,xn);
        ke = ke + 1;
    end
    xm = xn_1 - 2^(n-2)*p*dt;
    fm = feval(fname,xm);
    ke = ke + 1;
    if fm >= fn_1,
        x0 = xn_1+(2^(n-2)*p*dt*(fn_2-fm))/(2*(fn_2-2*fn_1+fm));
    else
        x0 = xm+(2^(n-2)*p*dt*(fn_1-fn))/(2*(fn_1-2*fm+fn));
    end
    er = 2^(n-2)*dt;
    dt = C*dt;
else
    x0 = x0 + dt*(f_1 - f1)/(2*(f_1 - 2*f0 + f1));
    er = dt;
    dt = C*dt;
end
k = k + 1;
end
% Apresentação de resultados
xs = x0;
fs = feval(fname,xs);
fprintf('Valor aproximado para o MINIMIZANTE: ');
fprintf(' %.6f\n', xs);
fprintf('Valor aproximado para o MÍNIMO: %.6f\n',fs);
fprintf('Número de ITERAÇÕES realizadas: %d\n',k);
fprintf('Número de AVALIAÇÕES da função objetivo: %d',ke);
fprintf(' %.6f\n');
end
```

Observação 3.10 O Algoritmo 3.3 também pode ser usado quando uma função é unimodal num intervalo $[a, b]$ de \mathbb{R} que contenha o ponto zero. Na verdade, é o que acontece com as funções dos problemas 3.5 e 3.6. Essas funções não são unimodais em \mathbb{R} . \square

Para finalizar o estudo do algoritmo de Davies, Swann e Campey, vamos aplicar o Algoritmo 3.3 à resolução de alguns problemas.

Problema 3.5 *Uma águia, ao efectuar um voo planado à procura de alimento, avistou uma lebre no fundo do vale de um parque natural. O fundo do vale é uma área plana. De imediato, a águia iniciou um voo picado, a grande velocidade, em direcção à presa,*

capturando-a em poucos segundos. Após a captura, transportou a lebre para o cimo de um penhasco, terminando aí o seu voo.

O momento da captura corresponde ao instante em que a águia atingiu, no seu voo, a distância mínima ao fundo do vale. Admita que a distância, h , em metros, a que a águia se encontra do fundo do vale, t segundos após o início do voo picado, é dada, aproximadamente, por

$$h(t) = -0.125t^4 + 2.5t^3 - 12.9t^2 - 1.1t + 94.8 \text{ com } t \in [0, 9.6].$$

Determine a distância da águia ao fundo do vale no momento da captura da lebre. Resolva o problema realizando duas iterações do algoritmo de Davies, Swann e Campey.

Resolução:

Para os efeitos da resolução deste problema, considera-se que $h = f$.

A função h não é unimodal em \mathbb{R} , mas é-o no intervalo $[0, 9.6]$ como mostra a Figura 3.8.

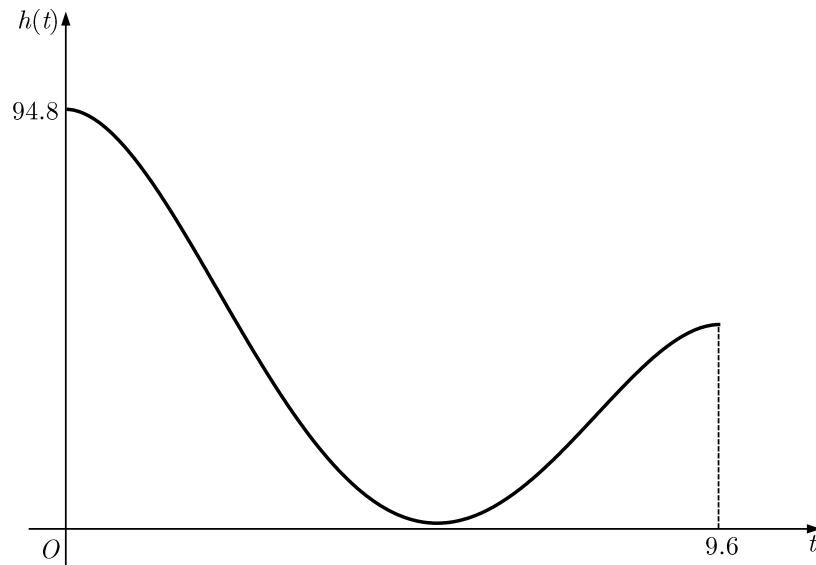


Figura 3.8: Função objetivo do Problema 3.5.

1.^a iteração

Passo 1

Vamos considerar o ponto inicial $x_{0,1} = 4$ (limite inferior do intervalo de incerteza inicial), $\delta_1 = 4 \times 0.1 = 0.4$ e $C = 0.1$. Não vamos precisar da tolerância de otimização, porque vamos realizar duas iterações. Seja $k = 0$.

Passo 2

$$k = 0 + 1 = 1,$$

$$x_{0,1} = 4,$$

$$x_{-1,k} = x_{0,k} - \delta_k \text{ e } x_{1,k} = x_{0,k} + \delta_k,$$

$$x_{-1,1} = x_{0,1} - \delta_1 = 4 - 0.4 = 3.6,$$

$$x_{1,1} = x_{0,1} + \delta_1 = 4 + 0.4 = 4.4,$$

$$f_{0,1} = f(4) = -0.125 \times 4^4 + 2.5 \times 4^3 - 12.9 \times 4^2 - 1.1 \times 4 + 94 = 12,$$

$$f_{1,1} = f(4.4) = -0.125 \times 4.4^4 + 2.5 \times 4.4^3 - 12.9 \times 4.4^2 - 1.1 \times 4.4 + 94 = 6.3248.$$

Passo 3

Como $f_{0,1} > f_{1,1}$, então $p = 1$.

Passo 4

$$\delta_1 = 0.4,$$

$$n = 1,$$

$$f_{n,k} = f(x_{n-1,k} + 2^{n-1}p\delta_k),$$

$$f_{1,1} = f(x_{0,1} + 2^{1-1} \times 1 \times \delta_1) = f(4 + 2^0 \times 0.4) = f(4.4),$$

$$f(4.4) = -0.125 \times 4.4^4 + 2.5 \times 4.4^3 - 12.9 \times 4.4^2 - 1.1 \times 4.4 + 94 = 6.3248,$$

$$n = 2,$$

$$f_{n,k} = f(x_{n-1,k} + 2^{n-1}p\delta_k),$$

$$f_{2,1} = f(x_{1,1} + 2^{2-1} \times 1 \times \delta_1) = f(4.4 + 2 \times 0.4) = f(5.2),$$

$$f(5.2) = -0.125 \times 5.2^4 + 2.5 \times 5.2^3 - 12.9 \times 5.2^2 - 1.1 \times 5.2 + 94 = 0.3888,$$

$$n = 3,$$

$$x_{2,k} = x_{1,k} + 2\delta_k,$$

$$x_{2,1} = x_{1,1} + 2\delta_1 = 4.4 + 2 \times 0.4 = 5.2,$$

$$f_{n,k} = f(x_{n-1,k} + 2^{n-1}p\delta_k),$$

$$f_{3,1} = f(x_{2,1} + 2^{3-1} \times 1 \times \delta_1) = f(5.2 + 2^2 \times 0.4) = f(6.8),$$

$$f(6.8) = -0.125 \times 6.8^4 + 2.5 \times 6.8^3 - 12.9 \times 6.8^2 - 1.1 \times 6.8 + 94 = 9.6368.$$

Passo 5

$$f_{m,k} = f(x_{n-1,k} + 2^{n-2}p\delta_k),$$

$$f_{m,1} = f(x_{2,1} + 2^{3-2} \times 1 \times \delta_1) = f(5.2 + 2 \times 0.4) = f(6),$$

$$f(6) = -0.125 \times 6^4 + 2.5 \times 6^3 - 12.9 \times 6^2 - 1.1 \times 6 + 94 = 1.8.$$

Passo 6

$$f_{m,k} = 1.8 \text{ e } f_{n-1,k} = 0.3888,$$

$$\text{Como } f_{m,k} > f_{n-1,k}, \text{ então: } x_{0,k+1} = x_{n-1,k} + \frac{2^{n-2}\delta_k(f_{n-2,k} - f_{m,k})}{2(f_{n-2,k} - 2f_{n-1,k} + f_{m,k})}.$$

$$x_{0,2} = x_{2,1} + \frac{2^1\delta_1(f_{1,1} - f_{m,1})}{2(f_{1,1} - 2f_{2,1} + f_{m,1})} \approx 5.2 + \frac{2 \times 0.4 \times (6.3248 - 1.8)}{2 \times (6.3248 - 2 \times 0.3888 + 1.8)} \approx 5.4476,$$

$$\delta_{k+1} = C \times \delta_k,$$

$$\delta_2 = 0.1 \times \delta_1 = 0.1 \times 0.4 = 0.04.$$

2.ª iteração

Passo 2

$$k = 1 + 1 = 2,$$

$$x_{0,2} = 5.4476 \text{ (ver iteração anterior),}$$

$$\delta_2 = 0.04 \text{ (ver iteração anterior),}$$

$$x_{-1,k} = x_{0,k} - \delta_k \text{ e } x_{1,k} = x_{0,k} + \delta_k,$$

$$x_{-1,2} = x_{0,2} - \delta_2 = 5.4476 - 0.04 = 5.4076,$$

$$x_{1,2} = x_{0,2} + \delta_1 = 5.4476 + 0.04 = 5.4876,$$

$$f_{0,2} = f(5.4476),$$

$$f(5.4476) = -0.125 \times 5.4476^4 + 2.5 \times 5.4476^3 - 12.9 \times 5.4476^2 - 1.1 \times 5.4476 + 94 \approx 0.0592,$$

$$f_{1,2} = f(5.4876),$$

$$f(5.4876) = -0.125 \times 5.4876^4 + 2.5 \times 5.4876^3 - 12.9 \times 5.4876^2 - 1.1 \times 5.4876 + 94 \approx 0.0720.$$

Passo 3

Como $f_{0,2}$ não é maior que $f_{1,2}$, então temos que calcular $f_{-1,2}$.

$$f_{-1,2} = f(5.4076),$$

$$f(5.4076) = -0.125 \times 5.4076^4 + 2.5 \times 5.4076^3 - 12.9 \times 5.4076^2 - 1.1 \times 5.4076 + 94 \approx 0.0647,$$

Como $f_{-1,2} \geq f_{0,2}$ e $f_{0,2} \leq f_{1,2}$, então devemos realizar o passo 7.

Passo 7

$$\delta_2 = 0.04,$$

$$x_{0,k+1} = x_{0,k} + \frac{\delta_k(f_{-1,k} - f_{1,k})}{2(f_{-1,k} - 2f_{0,k} + f_{1,k})},$$

$$x_{0,3} = x_{0,2} + \frac{\delta_2(f_{-1,2} - f_{1,2})}{2(f_{-1,2} - 2f_{0,2} + f_{1,2})} \approx 5.4476 + \frac{0.04 \times (0.0647 - 0.0720)}{2 \times (0.0647 - 2 \times 0.0592 + 0.0720)},$$

$$x_{0,3} \approx 5.4396.$$

Passo 8

$$x^* \approx x_{0,3} \approx 5.4396,$$

$$h(5.4396) \approx -0.125 \times 5.4396^4 + 2.5 \times 5.4396^3 - 12.9 \times 5.4396^2 - 1.1 \times 5.4396 + 94.8,$$

$$h(5.4396) \approx 0.0589.$$

No instante em que a águia capturou a sua presa ($t^* \approx 5.44$ segundos), encontrava-se a cerca de 5.9 cm do fundo do vale.

Problema 3.6 A altitude z , em metros, de objeto de massa m , em quilogramas, projetado para cima a partir da superfície da Terra, sujeito a resistência linear de coeficiente c , em Kg/s, com velocidade inicial v_0 , em m/s, e altitude inicial z_0 , em metros, é dada em função do tempo t , em segundos, por

$$z = z_0 + \frac{m}{c} \left(v_0 + \frac{mg}{c} \right) (1 - e^{-\frac{c}{m}t}) - \frac{mg}{c}t,$$

desde que $z > 0$. g representa a aceleração da gravidade.

Admita que é projetado para cima um objeto de 80 Kg de massa, a uma altitude inicial de 100 m, com velocidade inicial de 55 m/s e sujeito a uma resistência linear de coeficiente 15 Kg/s. Considere que a aceleração da gravidade é 9.8 m/s².

Qual é a altura máxima atingida pelo objeto nos 10 primeiros segundos após ter sido projetado? Em que instante ocorreu? Resolva o problema usando o algoritmo de Davies, Swann e Campey.

Resolução:

Considerando os dados, a função objetivo fica definida por

$$z(t) = 100 + \frac{80}{15} \times \left(55 + \frac{80 \times 9.8}{15} \right) \times (1 - e^{-\frac{15}{80}t}) - \frac{80 \times 9.8}{15}t,$$

$$z(t) = 100 + \frac{25744}{45} \times (1 - e^{-0.1875t}) - \frac{784}{15}t$$

Como sugere a Figura 3.9, a função z é unimodal no intervalo $[0, 10]$. Para resolver o problema, vamos minimizar $-z(t)$, aplicando, a este problema, o programa `davies_swann_campey.m`. Para tal, no editor do MATLAB, escrevemos

```
davies_swann_campey(inline('(100+(25744/45)*(1-exp(-0.1875*t))-784*t/15)')', 't'), 0, 0.05, 0.1, 10^-6)
```

Abaixo apresenta-se o resultado da aplicação da função `davies_swann_campey`.

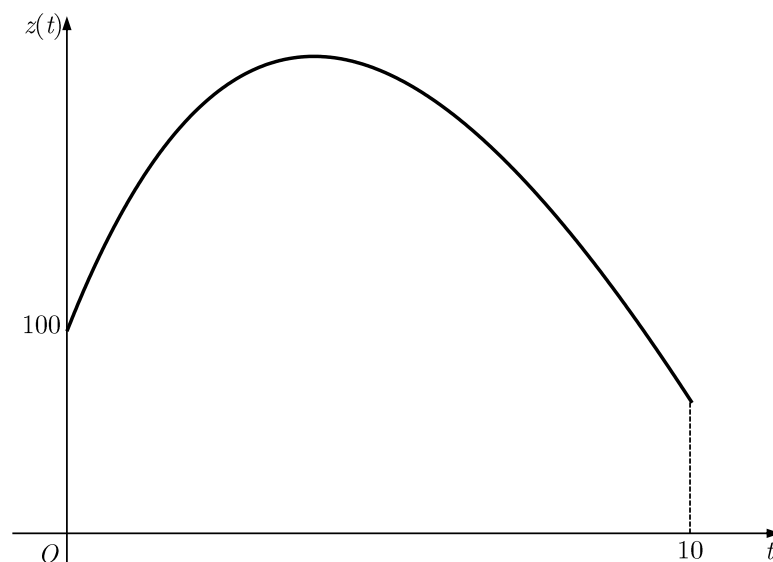


Figura 3.9: Função objetivo do Problema 3.6.

```
PROGRAMA davies_swann_campey.m
Valor aproximado para o MINIMIZANTE: 3.834449
Valor aproximado para o MÍNIMO: -192.919483
Número de ITERAÇÕES realizadas: 8
Número de AVALIAÇÕES da função objetivo: 85
```

A altura máxima atingida pelo objeto é aproximadamente igual a 192.9 m e ocorreu cerca de 3.8 s após a sua projeção.

Observação 3.11 A principal força que afeta o movimento de um projétil, para além da força da gravidade, é a força de resistência do ar. Trata-se de uma força retardadora e atua no sentido contrário à velocidade do projétil. Para projéteis que se movem no ar a velocidades relativamente baixas, a força de resistência do ar é aproximadamente proporcional ao módulo da velocidade, sendo que a constante de proporcionalidade c recebe a denominação de coeficiente de resistência linear.

O coeficiente de resistência linear c é uma constante positiva que depende da textura e da forma do objeto. Para objetos ásperos e de formas irregulares, o coeficiente pode ser maior que 1, enquanto que para objetos macios e de formas regulares é muito menor que 1. \square

Capítulo 4

Busca Linear Inexata

Em alguns algoritmos de otimização multidimensional, a maior parte do esforço computacional é gasto na avaliação da função objetivo e do seu gradiente na execução de buscas lineares. Consequentemente, a quantidade de cálculos necessários tende a depender da eficiência e precisão das buscas lineares utilizadas. Se for necessária uma busca linear de alta precisão, a quantidade de cálculos será grande e se as buscas lineares inexatas não afetarem a convergência de um algoritmo, uma pequena quantidade de cálculos pode ser suficiente.

Muitos métodos de otimização foram criados para serem bastante tolerantes a imprecisão da busca linear e, por esta razão, as buscas lineares inexatas são normalmente utilizadas nestes métodos.

O desenvolvimento deste capítulo será baseado nos autores [1, Antoniou e Lou], [14, 15, Fletcher], [6, Chong e Žak] e [25, Nocedal e Wright].

4.1 Condições de Goldstein

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função cujo mínimo se pretende determinar. A fórmula iteradora para a determinação do minimizante de f é da forma $x_{k+1} = x_k + \alpha_k d_k$, onde x_0 é um ponto inicial dado e α_k é um parâmetro não negativo que é escolhido para minimizar a função real de variável real ϕ_k , definida por $\phi_k(\alpha) = f(x_k + \alpha d_k)$. O vetor d_k define a direção de busca e α_k é o comprimento do passo. A escolha de α_k envolve, portanto, minimização univariada e garante que, sob determinadas condições, $f(x_{k+1}) < f(x_k)$. Para um certo valor positivo de α , consideremos que a função ϕ_k tem um único mínimo.

Seja $\nabla f(x_k) = g_k$. A aproximação linear da série de Taylor para $f(x_{k+1})$ é

$$f(x_{k+1}) = f(x_k) + g_k^T d_k \alpha, \quad (4.1)$$

onde

$$g_k^T d_k = \left. \frac{df(x_k + \alpha d_k)}{d\alpha} \right|_{\alpha=0}.$$

A Equação (4.1) define a reta a representada na Figura 4.1, enquanto que a reta b , nessa figura, para $0 \leq \rho < \frac{1}{2}$, é definida pela equação

$$f(x_{k+1}) = f(x_k) + \rho g_k^T d_k \alpha. \quad (4.2)$$

O declive desta reta varia entre 0 e $\frac{1}{2}g_k^T d_k$, dependendo do valor de ρ , como mostra a região a vermelho na Figura 4.1. Por outro lado, a equação

$$f(x_{k+1}) = f(x_k) + (1 - \rho)g_k^T d_k \alpha \quad (4.3)$$

representa a reta c na Figura 4.1, cujo declive varia entre $\frac{1}{2}g_k^T d_k$ e $g_k^T d_k$, como mostra a região verde da Figura 4.1. O ângulo entre as retas b e c , designado por θ na Figura 4.2, sendo m_b o declive da reta b e m_c o declive da reta c , é tal que

$$\tan \theta = \left| \frac{m_c - m_b}{1 + m_b m_c} \right|. \quad (4.4)$$

Então,

$$\tan \theta = \left| \frac{(1 - \rho)g_k^T d_k - \rho g_k^T d_k}{1 + \rho g_k^T d_k (1 - \rho)g_k^T d_k} \right|, \quad \text{por (4.4) e porque } m_b = \rho g_k^T d_k \text{ e } m_c = (1 - \rho)g_k^T d_k,$$

$$\tan \theta = \frac{\rho g_k^T d_k - (1 - \rho)g_k^T d_k}{1 + \rho g_k^T d_k (1 - \rho)g_k^T d_k}, \quad \text{pela definição de módulo e } (1 - \rho)g_k^T d_k < \rho g_k^T d_k < 0,$$

$$\tan \theta = \frac{-(1 - 2\rho)g_k^T d_k}{1 + \rho(1 - \rho)(g_k^T d_k)^2}, \quad \text{pelas propriedades distributiva e associativa.} \quad (4.5)$$

Da Igualdade (4.5) resulta que o ângulo das retas b e c , ilustrado na Figura 4.2 é o ângulo agudo tal que

$$\theta = \tan^{-1} \left[\frac{-(1 - 2\rho)g_k^T d_k}{1 + \rho(1 - \rho)(g_k^T d_k)^2} \right]. \quad (4.6)$$

Ao fixarmos ρ no intervalo $\left[0, \frac{1}{2}\right]$, existem dois valores de α , α_1 e α_2 , para os quais as retas c e b interseitam a curva $f(x_{k+1})$, como se mostra na Figura 4.2.

Consideremos que α_0 é uma estimativa para o valor de α que minimiza $f(x_k + \alpha d_k)$. Se, para $\alpha_k = \alpha_0$,

$$f(x_{k+1}) \leq f(x_k) + \rho g_k^T d_k \alpha_k \quad (4.7)$$

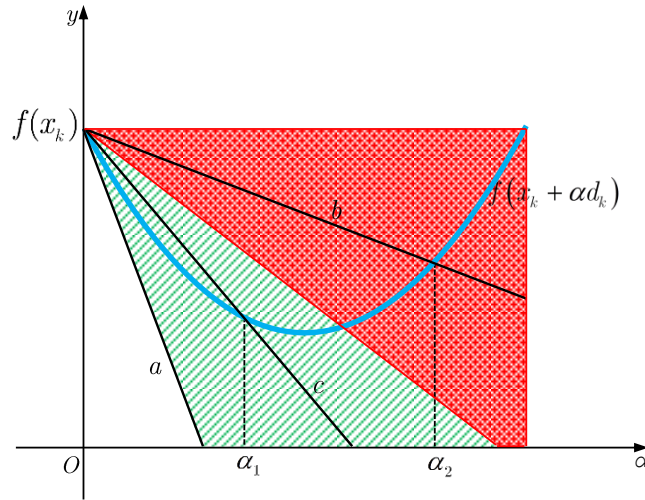


Figura 4.1: Teste de Goldstein.

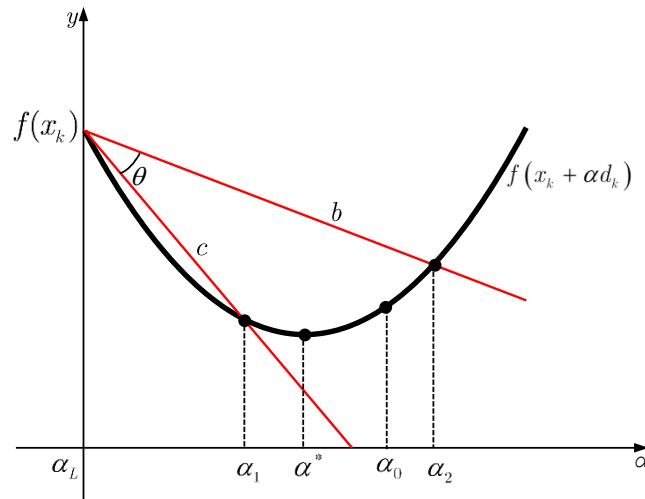


Figura 4.2: Teste de Goldstein satisfeito.

e

$$f(x_{k+1}) \geq f(x_k) + (1 - \rho)g_k^T d_k \alpha_k, \quad (4.8)$$

então α_0 pode ser considerada uma estimativa aceitável para α^* . Como sugere a Figura 4.2, nestas condições $\alpha_1 \leq \alpha_0 \leq \alpha_2$.

As desigualdades (4.7) e (4.8) constituem a base dos métodos de busca linear inexata, sendo conhecidas como *condições de Goldstein*. Nestes métodos, a estimativa α_0 é gerada com base na informação disponível e considerando a satisfação das condições de Goldstein. Se se verificarem as duas condições, a redução de $f(x_{k+1})$ é considerada aceitável e dá-se por terminado o processo. Caso contrário, se alguma das condições não se verificar, a redução de $f(x_{k+1})$ é insuficiente, devendo ser encontrada uma estimativa melhor para

α^* . Designemos por $\check{\alpha}_0$ essa estimativa. Se for a condição (4.7) a não se verificar, então $\alpha_0 > \alpha_2$, como se mostra na Figura 4.3, e uma vez que $\alpha_L < \alpha^* < \alpha_0$, a nova estimativa $\check{\alpha}_0$ pode ser calculada usando interpolação. Caso não se verifique a condição (4.8), $\alpha_0 < \alpha_1$ (ver Figura 4.4) e dado que é provável que α_0 esteja no intervalo $]\alpha_L, \alpha^*[$, $\check{\alpha}_0$ pode ser determinado usando extrapolação.

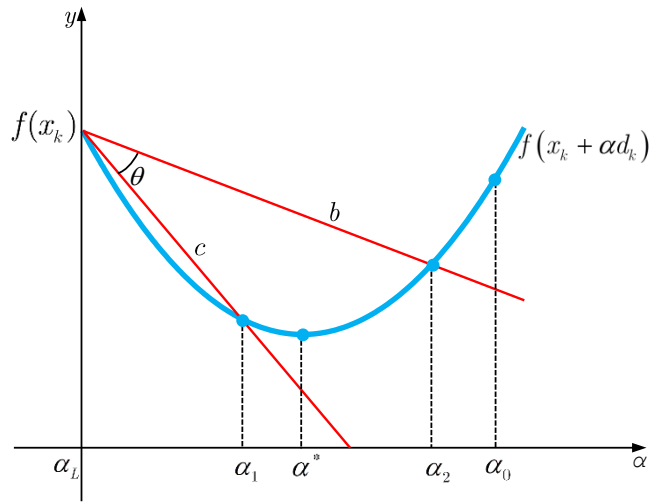


Figura 4.3: Teste de Goldstein não satisfeito. Caso $\alpha_0 > \alpha_2$.

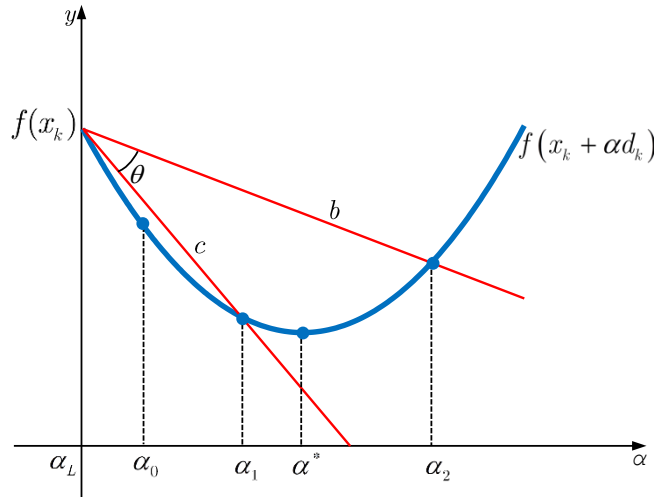


Figura 4.4: Teste de Goldstein não satisfeito. Caso $\alpha_0 < \alpha_1$.

Se o valor de $f(x_k + \alpha d_k)$ e da sua derivada em ordem a α forem conhecidos para $\alpha = \alpha_L$ e $\alpha = \alpha_0$, então para $\alpha_0 > \alpha_2$, uma boa estimativa para $\check{\alpha}_0$ pode ser obtida usando

a fórmula de interpolação quadrática em dois pontos (Fórmula (3.14), pág. 94). Fazendo, nessa fórmula, $x_1 = \alpha_L$, $x_2 = \alpha_0$ e $\bar{x} = \check{\alpha}_0$, obtém-se a fórmula interpoladora

$$\check{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 f'_L}{2[f_L - f_0 + (\alpha_0 - \alpha_L)f'_L]}. \quad (4.9)$$

Se em vez de se verificar $\alpha_0 > \alpha_2$, ocorrer $\alpha_0 < \alpha_1$, para obter a estimativa deve usar-se a fórmula de extrapolação

$$\check{\alpha}_0 = \alpha_0 + \frac{(\alpha_0 - \alpha_L)f'_0}{f'_L - f'_0}, \quad (4.10)$$

que se obtém da Fórmula (3.17) (pág. 96) fazendo $x_1 = \alpha_L$, $x_2 = \alpha_0$ e $\bar{x} = \check{\alpha}_0$. Note-se que nas igualdades (4.9) e (4.10)

$$f_L = f(x_k + \alpha_L d_k),$$

$$f'_L = f'(x_k + \alpha_L d_k) = g(x_k + \alpha_L d_k)^T d_k,$$

$$f_0 = f(x_k + \alpha_0 d_k),$$

$$f'_0 = f'(x_k + \alpha_0 d_k) = g(x_k + \alpha_0 d_k)^T d_k.$$

Repetindo a aplicação do procedimento exposto, eventualmente, podemos alcançar um valor de $\check{\alpha}_0$ tal que $\alpha_1 < \check{\alpha}_0 < \alpha_2$, o que permite terminar a busca linear inexata.

Observação 4.12 Nas condições de Goldstein, a condição (4.7) é a condição suficiente de decréscimo, enquanto que a condição (4.8) controla o comprimento do passo α_k . \square

Observação 4.13 As condições de Goldstein não as únicas que garantem uma redução suficiente do valor da função objetivo, de iteração para iteração, na base de que o passo α_k não deve ser nem demasiado grande nem demasiado pequeno. Referiremos de forma breve as condições propostas por Larry Armijo em 1966 e por Philip Wolfe em 1969.

Sejam c_1 , c_2 e γ três constantes tais que $c_1 \in]0, 1[$, $c_2 \in]c_1, 1[$ e $\gamma > 1$. As condições de Armijo garantem que o passo α_k não é demasiado grande exigindo que

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k. \quad (4.11)$$

Para assegurar que α_k não é demasiado pequeno essas condições estabelecem que

$$f(x_k + \gamma \alpha_k d_k) \geq f(x_k) + c_1 \gamma \alpha_k \nabla f(x_k)^T d_k. \quad (4.12)$$

A condição suficiente de decréscimo (condição (4.11)) não é bastante para assegurar que um algoritmo faça progressos razoáveis, porque é satisfeita por qualquer valor de α_k suficientemente pequeno. Para descartar a possibilidade de ocorrerem pequenos passos

inaceitáveis, em vez da condição (4.12), proposta por Armijo, Wolfe propôs a *condição de curvatura*, a qual requer que α_k satisfaça a desigualdade

$$\nabla f(x_k + \alpha d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k. \quad (4.13)$$

Deste modo, as condições (4.11) e (4.13) são conhecidas como *condições de Wolfe*.

Tal como as condições de Goldstein, as condições de Wolfe também garantem que o comprimento do passo α_k atinja um decrescimento suficiente, enquanto impedem que α_k seja demasiado pequeno.

Uma desvantagem das condições de Goldstein, no confronto direto com as condições de Wolfe, tem a ver com o facto de, na Desigualdade (4.7), se poderem excluir todos os minimizantes de ϕ_k .

O passo α_k pode satisfazer as condições de Wolfe sem estar particularmente próximo do minimizante de ϕ_k . Pelo que se impõe uma modificação da condição de curvatura para forçar α_k a estar pelo menos numa vizinhança ampla de um minimizante local ou de um ponto de estacionaridade de ϕ_k . Deste modo, obtêm-se as *condições fortes de Wolfe*, as quais exigem que α_k verifique as desigualdades

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T d_k \quad (4.14)$$

e

$$|\nabla f(x_k + \alpha d_k)^T d_k| \geq c_2 |\nabla f(x_k)^T d_k|. \quad (4.15)$$

□

4.2 Algoritmo de Busca Linear Inexata de Fletcher

O resultado seguinte é muito útil, pois é relativo à aplicação das condições de Goldstein numa busca linear inexata.

Teorema 4.1 (convergência de uma busca linear inexata) *Se*

- $f(x_k)$ for limitada inferiormente,
- g_k é uniformemente contínua no conjunto $\{x : f(x) < f(x_0)\}$,
- para todo k , as direções d_k não são ortogonais a $-g_k$,

então um algoritmo de descida usando busca linear inexata baseada nas condições de Goldstein converge para um ponto de estacionaridade, quando $k \rightarrow +\infty$.

A prova deste teorema é apresentada por [15, Fletcher, pág. 30]. O Teorema 4.1 é importante na medida que garante que as imprecisões devidas pela inexatidão da busca linear não são determinantes para a convergência.

Normalmente, as duas primeiras condições do Teorema 4.1 são satisfeitas, mas a terceira não. Este problema pode ser evitado na prática alterando a direção d_k . Por exemplo, se θ_k for o ângulo entre d_k e $-g_k$, sendo $\theta_k = \frac{-g_k^T d_k}{\|g_k\| \|d_k\|} = \frac{\pi}{2}$ (radianos), então d_k pode ser ligeiramente modificado para garantir que $\theta_k = \frac{\pi}{2} - \mu$, com $\mu > 0$.

Às vezes, as condições de Goldstein conduzem-nos à situação ilustrada na Figura 4.5, onde $\alpha^* \notin [\alpha_1, \alpha_2]$. Evidentemente, nesse caso o valor de α_0 no intervalo $[\alpha^*, \alpha_1]$ não permitirá terminar a busca linear, mesmo que a redução de $f(x_k)$ seja maior que qualquer α_0 no intervalo $[\alpha_1, \alpha_2]$. Embora o problema não seja grave, uma vez que a convergência está garantida pelo Teorema 4.1, a quantidade de cálculos pode ser aumentada. O problema pode ser resolvido substituindo a segunda condição de Goldstein, ou seja, a Desigualdade (4.8), pela condição

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad (4.16)$$

onde $0 < \sigma < 1$ e $\sigma \geq \rho$. Esta modificação foi proposta por Fletcher [14] e é ilustrada na Figura 4.6. O escalar $g_k^T d_k$ é a derivada de $f(x_k + \alpha d_k)$ para $\alpha = 0$ e, dado que $0 < \sigma < 1$, $\sigma g_k^T d_k$ é a derivada de $f(x_k + \alpha d_k)$ para um valor $\alpha = \alpha_1$ tal que $\alpha_1 < \alpha^*$. Agora, se a Desigualdade (4.16) se verificar em algum ponto $x_{k+1} = x_k + \alpha_0 d_k$, então o declive de $f(x_k + \alpha d_k)$ para $\alpha = \alpha_0$ é menos negativo (mais positivo) do que o declive para $\alpha = \alpha_1$ e, conseqüentemente, conclui-se que $\alpha_1 \leq \alpha_0$. Se se verificar também a Desigualdade (4.7) (primeira condição de Goldstein), ter-se-á $\alpha_1 < \alpha^* < \alpha_2$ ou $\alpha_1 < \alpha_0 < \alpha_2$, como se mostra na Figura 4.6.

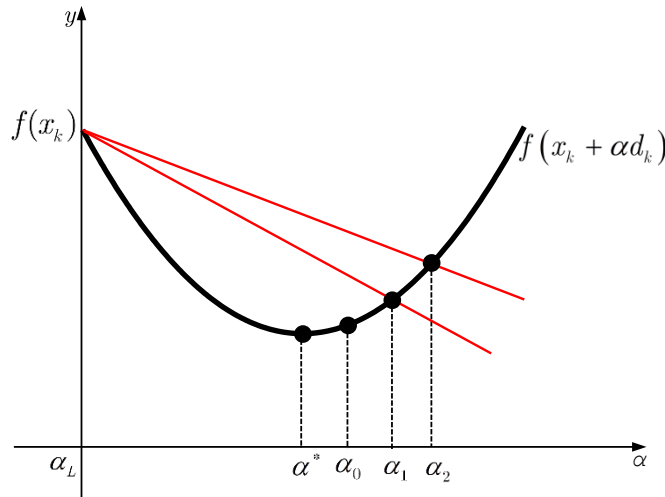
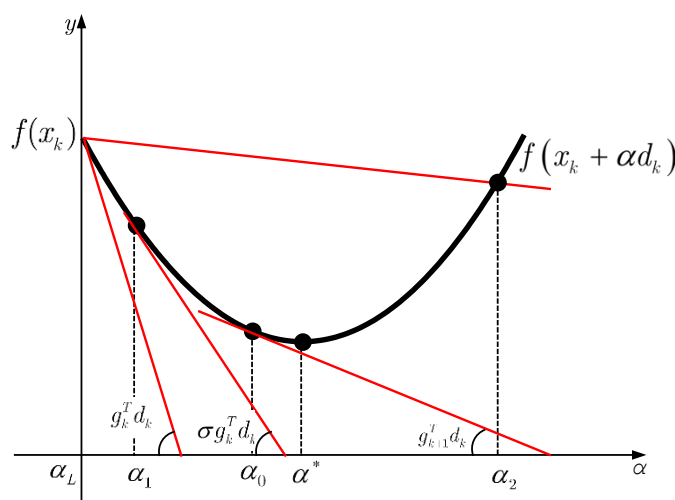


Figura 4.5: Teste de Goldstein não satisfeito com $\alpha^* < \alpha_1$.



No passo 7(e), os dados conhecidos que podem ser utilizados na iteração seguinte são guardados, isto é, α_0 , f_0 e f'_0 passam a ser α_L , f_L e f'_L , respetivamente. Isso mantém a quantidade de cálculos no mínimo.

Note-se que a primeira condição de Goldstein (Igualdade (4.7)) é modificada no passo 5 para ter em conta o facto de que α_L assume um valor maior do que zero quando a extrapolação é aplicada pelo menos uma vez.

A aplicação do Algoritmo 4.1 envolve uma quantidade considerável de cálculos, os quais justificam o recurso ao computador e à consequente implementação desse algoritmo numa linguagem de programação — o MATLAB neste caso. Faremos uso da rotina *busca_linear_inexata.m*, a qual foi adaptada da função *inex_lsearch.m* proposta por [1, Antoniou e Lou]. Apresentamos, então, a seguir, a função *busca_linear_inexata.m*.

```
% Programa: busca_linear_inexata.m
% Título: Busca Linear Inexata
% Descrição: Implementação do algoritmo de busca linear inexata de Fletcher
% Dados de entrada:
%   xk: ponto inicial
%   s: direção de busca
%   F: função objetivo a ser minimizada ao longo da direção s
%   G: gradiente da função objetivo F
%   p1: parâmetros internos que são necessários para a implementação da
%       busca linear, independentemente da aplicação à mão.
%       É um vetor e pode ser uma combinação de
%       vários parâmetros internos (por exemplo, 'rho = 0.25; sigma=0.5').
%       Habitualmente p1's incluem:                valor padrão
%       'rho=' define o suporte direito                0.1
%       'sigma=' define o suporte esquerdo (sigma >= rho) 0.1
%       'tau=' define passo mínimo para seccionamento 0.1
%       'chi='                                         0.75
%   p2: vetor de parametros definidos pelo utilizador. p2 deve ser um
%       vetor com todas as componentes numericamente especificadas.
%       As componentes de p2 devem aparece na mesma ordem que aparecem
%       na função F e no gradiente G. Por exemplo, se p2=[a b],
%       então F.m and G.m devem ser funções da forma z = F(x,p2)
%       e z = G(x,p2)
% Dados de saída:
%   z: valor aceitável de alfa
% Exemplo 1:
% Realização da busca linear inexata usando a função de Himmelblau
%   f(x1,x2) = (x1^2 + x2 - 11)^2 + (x1 + x2^2 - 7)^2
% começando pelo ponto xk = [6 6]' ao longo da direção de busca s = [-1 -1]'
% usando os parâmetros padrão.
% Solução:
% Executar o comando
%   a1 = busca_linear_inexata([6 6]',[-1 -1]', 'f_himm', 'g_himm')
% Exemplo 2:
% Realização da busca linear inexata usando a função de Himmelblau
% começando pelo ponto xk = [6 6]' ao longo da direção de busca s = [-1 -1]'
% com sigma = 0.5.
```

```
% Solução:
% Executar o comando
% a2 = busca_linear_inexata([6 6],[-1 -1'],'f_himm','g_himm','sigma = 0.5')
% Exemplo 3:
% Realização da busca linear inexata usando a função de Himmelblau
% parametrizada
% f(x1,x2,a,b) = (x1^2 + x2 - a^2)^2 + (x1 + x2^2 - b^2)^2
% começando pelo ponto xk = [6 6]' ao longo da direção de busca s = [-1 -1]',
% com parâmetros a = 3.2 e b = 2.6.
% Solução:
% Executar o comando
% a3 = busca_linear_inexata([6 6],[-1 -1'],'f_himm_p','g_himm_p',[3.2 2.6])
% Notas:
% 1. Comando
% z = busca_linear_inexata(xk,s,F,G,p1,p2)
% adiciona uma nova função busca_linear_inexata ao vocabulário do MATLAB.
% 2. Não use ponto e vírgula nos comandos
% a1 = busca_linear_inexata([6 6],[-1 -1'],'f_himm','g_himm')
% a2 = busca_linear_inexata([6 6],[-1 -1'],'f_himm','g_himm','sigma = 0.5')
% a3 = busca_linear_inexata([6 6],[-1 -1'],'f_himm_p','g_himm_p',[3.2 2.6])
% caso contrário o valor aceitável de alfa não será apresentado.
% 3. f_himm and g_himm são funções definidas pelo utilizador implementadas
% em ficheiros MATLAB f_himm.m e g_himm.m, respetivamente, são usadas
% para avaliar a função de Himmelblau e o seu gradiente. Analogamente,
% f_himm_p and g_himm_p são funções definidas pelo utilizados para a
% função de Himmelblau e para seu o gradiente.
%-----
function busca_linear_inexata(xk,s,F,G,p1,p2)
disp(' ');
disp('PROGRAMA busca_linear_inexata.m');
k = 0;
m = 0;
tau = 0.1;
chi = 0.75;
rho = 0.1;
sigma = 0.1;
mhat = 400;
epsilon = 1e-10;
xk = xk(:);
s = s(:);
parameterstring = '';
% avaliação dos parâmetros dados:
if nargin > 4,
    if isstr(p1),
        eval([p1 ' ','']);
    else
        parameterstring = ',p1';
    end
end
if nargin > 5,
    if isstr(p2),
```

```

        eval([p2 ' ','']);
    else
        parameterstring = ',p2';
    end
end
% Cálculo de f0 e g0
eval(['f0 = ' F '(xk' parameterstring ');']);
eval(['gk = ' G '(xk' parameterstring ');']);
m = m+2;
deltaf0 = f0;
% passo 2: Inicialização da busca linear
dk = s;
aL = 0;
aU = 1e308;
fL = f0;
dfL = gk'*dk;
if abs(dfL) > epsilon,
    a0 = -2*deltaf0/dfL;
else
    a0 = 1;
end
if ((a0 <= 1e-9)|(a0 > 1)),
    a0 = 1;
end
%passo 3
while 1,
    deltak = a0*dk;
    eval(['f0 = ' F '(xk+deltak' parameterstring ');']);
    m = m + 1;
%passo 4
    if ((f0 > (fL + rho*(a0 - aL)*dfL)) & (abs(fL - f0) > epsilon) & (m < mhat))
        if (a0 < aU)
            aU = a0;
        end
        % cálculo de a0hat
        a0hat = aL + ((a0 - aL)^2*dfL)/(2*(fL - f0 + (a0 - aL)*dfL));
        a0Lhat = aL + tau*(aU - aL);
        if (a0hat < a0Lhat)
            a0hat = a0Lhat;
        end
        a0Uhat = aU - tau*(aU - aL);
        if (a0hat > a0Uhat)
            a0hat = a0Uhat;
        end
        a0 = a0hat;
    else
        eval(['gtemp = ' G '(xk+a0*dk' parameterstring ');']);
        df0 = gtemp'*dk;
        m = m + 1;
        % passo 6
        if (((df0 < sigma*dfL) & (abs(fL - f0) > epsilon) & (m < mhat) & (dfL ~= df0)))

```

```

    deltaa0 = (a0 - aL)*df0/(dfL - df0);
    if (deltaa0 <= 0)
        a0hat = 2*a0;
    else
        a0hat = a0 + deltaa0;
    end
    a0Uhat = a0 + chi*(aU - a0);
    if (a0hat > a0Uhat)
        a0hat = a0Uhat;
    end
    aL = a0;
    a0 = a0hat;
    fL = f0;
    dfL = df0;
else
    break;
end
end
end % while 1
if a0 < 1e-5,
    z = 1e-5;
else
    z = a0;
fprintf('Valor aceitável para o MINIMIZANTE alfa: ');
fprintf(' %.6f', z);
end

```

Para concluir este capítulo, vamos aplicar o Algoritmo 4.1 à resolução de um problema.

Problema 4.1 *Seja $\mathcal{D} = \{(x_1, x_2) \in \mathbb{R}^2 : -\pi \leq x_1 \leq \pi \wedge -\pi \leq x_2 \leq \pi\}$.*

Considere a função, de domínio \mathcal{D} , definida por

$$f(x_1, x_2) = 0.7x_1^4 - 8x_1^2 + 6x_2^2 + \cos(x_1x_2) - 8x_1.$$

Utilize a busca linear inexata de Fletcher para atualizar o ponto x_0 ao longo da direção de busca d_0 . Resolva o problema

$$\begin{aligned} &\text{minimizar } f(x_0 + \alpha d_0), \\ &\alpha \geq 0 \end{aligned}$$

sabendo que

$$(a) \quad x_0 = \begin{bmatrix} -\pi \\ \pi \end{bmatrix} \quad e \quad d_0 = \begin{bmatrix} 1 \\ -1.3 \end{bmatrix}.$$

$$(b) \ x_0 = \begin{bmatrix} -\pi \\ \pi \end{bmatrix} \quad e \quad d_0 = \begin{bmatrix} 1 \\ -1.1 \end{bmatrix}.$$

Resolução:

Executando no MATLAB as instruções

```
syms x1 x2
f=0.7*x1^4-8*x1^2+6*x2^2+cos(x1*x2)-8*x1;
dx1=diff(f,x1) %derivada em ordem a x1
dx2=diff(f,x2) %derivada em ordem a x2
```

obtemos as derivadas parciais de f

```
dx1 =
(14*x1^3)/5 - x2*sin(x1*x2) - 16*x1 - 8

dx2 =
12*x2 - x1*sin(x1*x2)
```

Para avaliar a função f e o seu gradiente ∇f , o programa busca_linear_inexata.m usará as seguintes funções:

```
%funcao.m
function z = funcao(x)
x1 = x(1);
x2 = x(2);
z = 0.7*x1^4-8*x1^2+6*x2^2+cos(x1*x2)-8*x1;
end

% gradiente.m
function z = gradiente(x)
x1 = x(1);
x2 = x(2);
z1 = (14*x1^3)/5 - x2*sin(x1*x2) - 16*x1 - 8;
z2 = 12*x2 - x1*sin(x1*x2);
z = [z1 z2]';
end
```

(a) No editor do MATLAB, executarmos o comando

```
busca_linear_inexata([-pi pi]',[1 -1.3]','funcao','gradiente').
```

Obtém-se a seguinte solução para o problema:

```
PROGRAMA busca_linear_inexata.m
```

```
Valor aceitável para o MINIMIZANTE alfa: 2.116264
```

Logo, sendo $x_1 = x_0 + \alpha^ d_0$ e $\alpha^* \approx 2.116264$, então*

$$x_1 = \begin{bmatrix} -\pi \\ \pi \end{bmatrix} + 2.116264 \times \begin{bmatrix} 1 \\ -1.3 \end{bmatrix} \approx \begin{bmatrix} -1.025329 \\ 0.390449 \end{bmatrix}$$

(b) No editor do MATLAB, executarmos o comando

```
busca_linear_inexata([-pi pi]',[1 -1.1]','funcao','gradiente').
```

A solução do problema é a seguinte:

```
PROGRAMA busca_linear_inexata.m
```

```
Valor aceitável para o MINIMIZANTE alfa: 4.595945
```

Deste modo, sendo $x_1 = x_0 + \alpha^ d_0$ e $\alpha^* \approx 4.595945$, então*

$$x_1 \approx \begin{bmatrix} -\pi \\ \pi \end{bmatrix} + 4.595945 \times \begin{bmatrix} 1 \\ -1.1 \end{bmatrix} = \begin{bmatrix} 1.454352 \\ -1.913947 \end{bmatrix}.$$

Algoritmo 4.1: Busca Linear Inexata de Fletcher.**Passo 1**

Entrada: x_k, d_k e calcular g_k . Inicializar os parâmetros do algoritmo ρ, σ, τ e χ .
Considerar $\alpha_L = 0$ e $\alpha_U = 10^{308}$ e a função objetivo f .

Passo 2

Calcular $f_L = f(x_k + \alpha_L d_k)$ e $f'_L = g(x_k + \alpha_L d_k)^T d_k$.

Passo 3

Estimar α_0 .

Passo 4

Calcular $f_0 = f(x_k + \alpha_0 d_k)$.

Passo 5 (Interpolação)

se $f_0 > f_L + \rho(\alpha_0 - \alpha_L)f'_L$ então

(a) se $\alpha_0 < \alpha_U$ então
| $\alpha_U \alpha_0$.

(b) Calcular $\check{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 f'_L}{2[f_L - f_0 + (\alpha_0 - \alpha_L)f'_L]}$ (Igualdade (4.9)).

(c) se $\check{\alpha}_0 < \alpha_L + \tau(\alpha_U - \alpha_L)$ então
| $\check{\alpha}_0 = \alpha_L + \tau(\alpha_U - \alpha_L)$.

(d) se $\check{\alpha}_0 > \alpha_U - \tau(\alpha_U - \alpha_L)$ então
| $\check{\alpha}_0 = \alpha_U - \tau(\alpha_U - \alpha_L)$.

(e) Considerar $\alpha_0 = \check{\alpha}_0$ e ir para o Passo 4.

Passo 6

Calcular $f'_0 = g(x_k + \alpha_0 d_k)^T d_k$.

Passo 7 (Extrapolação)

se $f'_0 < \sigma f'_L$ então

(a) Calcular $\Delta\alpha_0 = \frac{(\alpha_0 - \alpha_L)f'_0}{f'_L - f'_0}$ (ver Igualdade (4.10)).

(b) se $\Delta\alpha_0 < \tau(\alpha_0 - \alpha_L)$ então
| $\Delta\alpha_0 = \tau(\alpha_0 - \alpha_L)$.

(c) se $\Delta\alpha_0 > \chi(\alpha_0 - \alpha_L)$ então
| $\Delta\alpha_0 = \chi(\alpha_0 - \alpha_L)$.

(d) Calcular $\check{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.

(e) Considerar $\alpha_L = \alpha_0, \alpha_0 = \check{\alpha}_0, f_L = f_0, f'_L = f'_0$ e ir para o Passo 4.

Passo 8

Resultado: α_0 e $f_0 = f(x_k + \alpha_0 d_k)$. Stop.

Conclusões

Nesta dissertação, foi apresentado um estudo integrado acerca de métodos de otimização local de uma função real, de uma variável real, definida num intervalo fechado. Fizemos a opção de não nos referirmos aos métodos *root-finding* aplicados à resolução da equação de estacionaridade $f'(x) = 0$. Estes métodos são bem conhecidos de todos e por questões de espaço não foram abrangidos no trabalho exposto.

Principais Objetivos Alcançados

Com a realização desta dissertação, fica claro que foram alcançados os seguintes propósitos:

- Apresentação rigorosa dos métodos clássicos de otimização baseados no Teorema de Fermat;
- Comparação da eficiência de métodos intervalares de eliminação;
- Estudo de propriedades dos números de Fibonacci que se revelam importantes neste contexto;
- Justificação de que, em certo sentido, o método de Fibonacci é o ideal (ótimo, portanto);
- Demonstração das razões da preferência do método da secção áurea em detrimento do método de Fibonacci, apesar deste ser mais eficiente;
- Exemplificação da operacionalização dos algoritmos dos métodos estudados;
- Utilização do MATLAB para implementação computacional de algoritmos e como ferramenta auxiliar na tomada de decisões.

Esta dissertação tem também uma vertente relativa à modelação matemática que julgamos atingida. Escolhemos muitos exemplos de situações reais que foram modeladas na forma de problemas de otimização univariada. Este aspeto do trabalho nunca perdeu de vista

algumas preocupações pedagógicas na exposição do texto, embora esse não fosse o foco principal. Nesse contexto, procurou-se apresentar gráficos que facilitassem a leitura. Por vezes usaram-se técnicas de resolução não recorrendo a ajudas computacionais. Outras vezes usaram-se algoritmos programados em MATLAB para obter as soluções.

Por fim, temos de referir que na pesquisa bibliográfica efetuada nos apercebemos de que a otimização univariada não é uma área tão restrita como normalmente se pensa. Convém também frisar que se continuam a publicar trabalhos científicos originais nas melhores revistas internacionais. Essas perspectivas abertas nas pesquisas bibliográficas vão mostrar, na secção seguinte, algumas pistas de trabalho adicional ainda dentro da otimização univariada.

Tópicos de Trabalho Complementar em Otimização Univariada

Problemas determinísticos resolvidos por métodos determinísticos

Otimização local Esta dissertação apresenta uma seleção de algoritmos para otimização de *funções unimodais* definidas em intervalos fechados de \mathbb{R} . Trata, portanto, de métodos de “*peak-seeking*” e de problemas de otimização local. Os problemas tratados são problemas determinísticos e são resolvidos por métodos determinísticos. Apenas alguns algoritmos exigem a diferenciabilidade da função porque passam por calcular derivadas da função objetivo. Na verdade, os algoritmos podem dividir-se em várias classes conforme a informação exigida:

- **Ordem zero:** Apenas usam informação acerca da função objetivo. Esta área é muitas vezes designada na literatura internacional por “*derivative-free optimization*”. Só faltou tratar alguns métodos baseados no método de Fibonacci (“*lattice methods*”, ver por exemplo [34, Wilde]). Trataram-se métodos intervalares assim como métodos interpoladores. Estes últimos, por vezes, já calculavam derivadas e estão inseridos na classe seguinte.
- **Ordem um:** Estes métodos calculam valores da função objetivo e da sua derivada. São designados por “*gradient methods*.” Muitos destes métodos procuram resolver a equação de estacionariedade $f'(x) = 0$. São também conhecidos por métodos “*root-finding*” ou “*zero-seeking*.” Incluem-se aqui os métodos de resolução de equações não lineares: bissecção, corda-falsa, secantes e interpolação inversa (algoritmo de Müller, por exemplo). Há métodos populares como o método de Brent que são combinações engenhosas dos métodos anteriores.
- **Ordem dois:** Calculam valores da função e das derivadas até à ordem dois. Estes métodos são a conhecida classe de “*Newton-like methods*.” Estes métodos tentam

resolver a equação $f'(x) = 0$ usando o método de Newton ou uma das suas variantes.

Otimização global Por outro lado, se a função for multimodal o problema deixa de ser de otimização local para se tornar num problema de otimização global. Note-se que, neste caso, pode não ser possível calcular o minimizante global. A existência de múltiplos minimizantes não permite, no caso geral, estabelecer condições de paragem que assegurem a aproximação de um minimizante local que seja global. Uma das pistas para trabalho adicional seria tratar métodos tipo “*branch and bound*” aplicados a otimização de funções Lipschitz contínuas de constante K conhecida. Ou seja, para casos em que a função objetivo cumpre a condição

$$|f(x_1) - f(x_2)| \leq K |x_1 - x_2|, \forall x_1, x_2 \in [a, b]. \quad (4.17)$$

O valor de K pode ser considerado como a taxa de variação máxima possível entre dois pontos quaisquer de $[a, b]$. Por vezes designa-se por “*slope bound*.” Se a função for diferenciável basta exigir que $|f'(x)| \leq K, \forall x \in [a, b]$ para que se cumpra a condição (4.17). Nestes casos, é possível calcular limites inferiores e superiores para a função na forma de funções afins por troços. Usar esses limites permite estabelecer algoritmos conhecidos: algoritmo minimax ou o método de Shubert-Pijavskij (ver [36, Zihigljavsky e Žilinskas]).

Toda esta área da optimização global de funções Lipschitz contínuas permitia prosseguir o trabalho desta dissertação. Continua a publicar-se nesta área de optimização univariada (ver, por exemplo, [16, Hansen], [28, Sergeyev, Stongin e Lera] e [29, Sergeyev e Lera]).

Temos de referir também que os problemas de optimização multidimensional podem ser formulados como problemas de optimização univariada. Na verdade, usando curvas de Peano podem colocar-se em correspondência biunívoca os pontos de um hipercubo de \mathbb{R}^n com os pontos do intervalo normalizado $[0, 1] \subset \mathbb{R}$. Trata-se de uma abordagem recente (ver [28, Sergeyev, Stongin e Lera]) que se tem designado por “*space-filling curves*.” Assim, os algoritmos de optimização univariada ganham outra importância. Esta abordagem é especialmente atraente se tivermos uma função “*black-box*”. Este tipo de funções aparece em muitas aplicações por duas vias: ou é definida por um algoritmo implementado em código-fonte fechado ou muito difícil de decifrar ou então necessita de ensaios laboratoriais para se calcular. Nesses casos, muitas vezes essa função demora muito tempo a ser calculada ou envolve custos laboratoriais. Muitas vezes não se sabem propriedades nenhuma dessa função além da continuidade. É neste contexto que os métodos baseados em “*space-filling curves*” são competitivos. Esta é outra pista de trabalho adicional.

Problemas determinísticos e métodos estocásticos

Em optimização local, os métodos determinísticos são adequados porque se conseguem estabelecer propriedades analíticas globais (convexidade, unimodalidade, continuidade à Lipschitz, etc.). No entanto, em optimização global a informação acerca da função é muitas

vezes fragmentada e difícil de integrar para definir procedimentos determinísticos. Assim, os métodos estocásticos (e estatísticos) são atrativos nestas situações.

Estes métodos baseiam-se todos em colocar pontos na região admissível recorrendo ao cálculo de números pseudo-aleatórios. Por vezes basta considerar um método determinístico com “*multi-start*” aleatório¹ para termos um método estocástico.

Os métodos estocásticos são tradicionalmente conhecidos por “*random search*” ou mesmo por métodos de Monte Carlo. Depois pode usar-se inferência estatística e os melhores resultados de convergência são aqueles que, sob certas circunstâncias, garantem *convergência com probabilidade um*.

Devemos referir que a otimização global é um dos campos mais difíceis das matemáticas computacionais. Assim, não é de espantar que tenham sido propostos várias abordagens para construir algoritmos. Algumas abordagens são chamadas heurísticas se bem que seja difícil classificar os métodos usando a dicotomia heurístico versus não heurístico. Muitas vezes essas heurísticas são suportadas por crenças (baseadas em massivos ensaios computacionais), nas quais se aceita que esses métodos conseguem soluções boas em tempo razoável. No entanto, alguns métodos heurísticos foram investigados matematicamente. Um exemplo característico é o “*simulated annealing*”, para o qual se pode garantir convergência para o minimizante global com probabilidade um, se forem executadas um número infinito de iterações. Convém frisar que, hoje em dia, já não há uma conotação negativa relativa às heurísticas mas muitas têm muito pouca teoria de suporte. A sua popularidade baseia-se na sua utilidade prática demonstrada em inúmeros ensaios e em muitos problemas diferentes. Deve ser sempre lembrado que estes métodos só são viáveis porque hoje em dia os computadores têm elevadas velocidades de processamento.

Há classes de heurísticas que se aplicam a uma grande diversidade de problemas. Chamam-se metaheurísticas dado que geram heurísticas que se adaptam bem na generalidade dos problemas de otimização. Por oposição, há heurísticas desenvolvidas para um problema de otimização específico tirando partido de propriedades especiais de uma certa formulação. O estudo das metaheurísticas tem evoluído nas últimas décadas e permitem hoje resolver problemas que antes eram difíceis ou impossíveis de solucionar. Algumas dessas heurísticas são (entre outras) o “*simulated annealing*”, a “*particle swarm*”, a “*ant colony*” e os “*genetic algorithms*”. Estes métodos modernos têm vantagens ao nível do tempo de implementação e da robustez. Deste modo, esta temática é uma excelente proposta para uma extensão deste trabalho. Para introdução a esses métodos ver, por exemplo, [33, Venkataraman] e [35, Yang].

¹ Nesses métodos em vez de escolher uma só aproximação inicial usa-se um conjunto de aproximações iniciais geradas aleatoriamente. De todos os resultados obtidos escolhe-se o melhor.

Problemas estocásticos e métodos estocásticos

A programação estocástica lida com situações em que alguns ou todos os parâmetros do problema de otimização são descritos por variáveis aleatórias e não por quantidades determinísticas. As fontes de variáveis aleatórias podem ser várias, de acordo com a natureza e o tipo de problema. Por exemplo, na concepção de estruturas de betão, a resistência do betão é uma variável aleatória, varia consideravelmente de amostra para amostra. Da mesma forma, no fabrico de aviões as cargas reais que atuam sobre estes dependem das condições atmosféricas prevalentes na altura do voo, as quais não podem ser previstas de forma precisa. Assim, as cargas devem ser tratados como variáveis aleatórias. Portanto, tem interesse prático o estudo da programação estocástica. Neste contexto, uma subárea interessante para estudo complementar seria a dos problemas em que a função objetivo é calculada com um erro experimental associado. Assim, a função é composta por uma componente determinística (sinal) e por uma componente aleatória (ruído). Podemos referir os métodos de Robbins-Monro para “*root-finding*” e de Kiefer-Wolfowitz para “*peek-seeking*.” Sobre este assunto sugerimos o livro de Douglass Wilde [34].

Apêndices

Apêndice A

Propriedades dos Números de Fibonacci

Neste apêndice serão estudadas algumas propriedades dos números de Fibonacci, nomeadamente aquelas que serão úteis para melhor compreender o método de Fibonacci discutido no Secção 2.4, assim como o método da secção áurea (ver Secção 2.5). Os autores [11, Sá Esteves] e [5, Chong e Žak] foram as referência escolhidas para o estudo apresentado a seguir.

A sucessão de Fibonacci é a sucessão definida recursivamente por

$$\begin{cases} F_0 = 1 \\ F_1 = 1 \\ F_n = F_{n-2} + F_{n-1}, \quad n \geq 2. \end{cases} \quad (\text{A.1})$$

Por convenção, neste trabalho, vamos assumir que $F_{-1} = 0$. Na tabela A.1 apresentam-se os onze primeiros número de Fibonacci.

| F_0 | F_1 | F_2 | F_3 | F_4 | F_5 | F_6 | F_7 | F_8 | F_9 | F_{10} |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 |

Tabela A.1: Números de Fibonacci.

A aplicação da sucessão de Fibonacci à otimização unidimensional gerou o método de Fibonacci. Este método usa os números de Fibonacci no cálculo dos pontos onde se determinarão as experiências da função objetivo.

A solução do problema

$$\text{minimizar} \quad (1 - \lambda_1) \times (1 - \lambda_2) \times \cdots \times (1 - \lambda_n)$$

sujeito a

$$\lambda_{k+1} = 1 - \frac{\lambda_k}{1 - \lambda_k}, \quad k = 1, 2, \dots, n-1 \quad (\text{A.2})$$

$$0 \leq \lambda_k \leq \frac{1}{2}, \quad k = 1, 2, \dots, n.$$

é importante para o método de Fibonacci, na medida em que garante a máxima contração do intervalo de incerteza. Vamos mostrar que $\lambda_k = 1 - \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$, $n \in \mathbb{N}$, é a sua única solução. Por uma questão de simplificação dos cálculos que faremos a seguir, vamos reescrever o Problema (A.2), considerando $1 - \lambda_k = R_k$ e $\psi(R_1, R_2, \dots, R_n) = R_1 \times R_2 \times \cdots \times R_n$. Obtemos, então, a seguinte formulação do referido problema:

$$\text{minimizar} \quad \psi(R_1, R_2, \dots, R_n)$$

sujeito a

$$R_{k+1} = \frac{1}{R_k} - 1, \quad k = 1, 2, \dots, n-1 \quad (\text{A.3})$$

$$\frac{1}{2} \leq R_k \leq 1, \quad k = 1, 2, \dots, n.$$

Portanto, mostraremos que a solução ótima é $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$, pelo que precisaremos das seguintes propriedades:

Lema A.1 *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci.*

Se R_1, R_2, R_3, \dots é uma sucessão de números reais tal que $R_{k+1} = \frac{1}{R_k} - 1$ e $\frac{1}{2} \leq R_k \leq 1$, $k = 1, 2, \dots$, então, para $k \geq 2$,

$$R_k = -\frac{F_{k-2} - F_{k-1} \times R_1}{F_{k-3} - F_{k-2} \times R_1}. \quad (\text{A.4})$$

Prova: A prova será feita com recurso ao método de indução.

Para $k = 2$, da Igualdade (A.4) obtém-se $R_2 = -\frac{F_0 - F_1 \times R_1}{F_{-1} - F_0 \times R_1} = -\frac{1 - 1 \times R_1}{0 - 1 \times R_1} = \frac{1}{R_1} - 1$. Esta igualdade é verdadeira, atendendo à hipótese.

Admitamos, agora, que para um certo valor m ($m \in \mathbb{N}$ e $m \geq 2$) se tem que

$$R_m = -\frac{F_{m-2} - F_{m-1} \times R_1}{F_{m-3} - F_{m-2} \times R_1} \quad (\text{hipótese de indução}).$$

Vamos mostrar que

$$R_{m+1} = -\frac{F_{m-1} - F_m \times R_1}{F_{m-2} - F_{m-1} \times R_1}.$$

Por hipótese, $R_{m+1} = \frac{1}{R_m} - 1$. Então, atendendo à hipótese de indução, tem-se

$$R_{m+1} = \frac{1}{-\frac{F_{m-2} - F_{m-1} \times R_1}{F_{m-3} - F_{m-2} \times R_1}} - 1$$

$$R_{m+1} = -\frac{F_{m-3} - F_{m-2} \times R_1}{F_{m-2} - F_{m-1} \times R_1} - 1$$

$$R_{m+1} = \frac{-(F_{m-3} - F_{m-2} \times R_1) - (F_{m-2} - F_{m-1} \times R_1)}{F_{m-2} - F_{m-1} \times R_1}$$

$$R_{m+1} = -\frac{F_{m-3} + F_{m-2} - (F_{m-2} + F_{m-1}) \times R_1}{F_{m-2} - F_{m-1} \times R_1}$$

$$R_{m+1} = -\frac{F_{m-1} - F_m \times R_1}{F_{m-2} - F_{m-1} \times R_1}.$$

Portanto, a Igualdade (A.4) é hereditária. Como adicionalmente se verifica para $k = 2$, conclui-se que é válida para todo número natural $k \geq 2$. \square

Lema A.2 *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci.*

Se R_1, R_2, R_3, \dots é uma sucessão de números reais tal que $R_{k+1} = \frac{1}{R_k} - 1$ e $\frac{1}{2} \leq R_k \leq 1$, $k = 1, 2, \dots$, então, para $k \geq 2$,

$$(-1)^k \times (F_{k-2} - F_{k-1}) \times R_1 > 0. \quad (\text{A.5})$$

Prova: Vamos recorrer ao método de indução para provar este Lema.

Seja $k = 2$. Da Desigualdade (A.5) resulta $(-1)^2 \times (F_0 - F_1) \times R_1 > 0$, ou seja $R_1 > 0$. Esta igualdade é verdadeira, atendendo à hipótese.

Vamos supor — hipótese de indução — que para um certo valor m ($m \in \mathbb{N}$ e $m \geq 2$) se tem que

$$(-1)^m \times (F_{m-2} - F_{m-1}) \times R_1 > 0.$$

Temos que mostrar que

$$(-1)^{m+1} \times (F_{m-1} - F_m) \times R_1 > 0.$$

Atendendo ao Lema A.1 obtém-se

$$R_{m+1} = -\frac{F_{m-1} - F_m \times R_1}{F_{m-2} - F_{m-1} \times R_1},$$

que equivalente a

$$R_{m+1} \times (F_{m-2} - F_{m-1} \times R_1) = -(F_{m-1} - F_m \times R_1).$$

Multiplicando os dois membros da última igualdade por $(-1)^k$, obtém-se

$$(-1)^k \times (F_{m-2} - F_{m-1} \times R_1) \times R_{m+1} = (-1)^{k+1} (F_{m-1} - F_m \times R_1).$$

Se tivermos em atenção a hipótese do Lema e a hipótese de indução, concluímos que $(-1)^{k+1} (F_{m-1} - F_m \times R_1) > 0$. O que garante a hereditariedade da propriedade e permite concluir a prova. \square

Lema A.3 *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci.*

Se R_1, R_2, R_3, \dots é uma sucessão de números reais tal que $R_{k+1} = \frac{1}{R_k} - 1$ e $\frac{1}{2} \leq R_k \leq 1$, $k = 1, 2, \dots$, então, para $k \geq 2$,

$$(-1)^{k+1} \times R_1 \geq (-1)^{k+1} \times \frac{F_k}{F_{k+1}}.$$

Prova: Uma vez que $R_{k+1} = \frac{1}{R_k} - 1$, conclui-se que $R_{k+1} \leq 1$. Substituindo, nesta última desigualdade, R_{k+1} por $-\frac{F_{k-1} - F_k \times R_1}{F_{k-2} - F_{k-1} \times R_1}$, atendendo ao Lema A.1, obtém-se

$$-\frac{F_{k-1} - F_k \times R_1}{F_{k-2} - F_{k-1} \times R_1} \leq 1.$$

Multiplicando os termos da fração, do membro esquerdo da última desigualdade, por $(-1)^k$, resulta

$$\frac{(-1)^{k+1} (F_{k-1} - F_k \times R_1)}{(-1)^k (F_{k-2} - F_{k-1} \times R_1)} \leq 1.$$

Porém, como $(-1)^k(F_{k-2} - F_{k-1} \times R_1) > 0$, de acordo com o Lema A.2, e atendendo à hipótese obtém-se

$$(-1)^{k+1}(F_{k-1} - F_k \times R_1) \leq (-1)^k(F_{k-2} - F_{k-1} \times R_1),$$

$$(-1)^{k+1}(F_{k-1} - F_k \times R_1) \leq (-1)^{k+1}(-F_{k-2} + F_{k-1} \times R_1),$$

$$(-1)^{k+1}(F_{k-1} + F_{k-2}) \leq (-1)^{k+1}(F_k + F_{k-1})R_1,$$

$$(-1)^{k+1} \times F_k \leq (-1)^{k+1} \times F_{k+1} \times R_1 \text{ (pela definição de números de Fibonacci),}$$

$$(-1)^{k+1} \times F_{k+1} \times R_1 \geq (-1)^{k+1} \times F_k,$$

$$(-1)^{k+1} \times R_1 \geq (-1)^{k+1} \times \frac{F_k}{F_{k+1}}.$$

O que permite concluir a prova. □

Lema A.4 *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci.*

Então, para $k \geq 2$,

$$F_{k-2} \times F_{k+1} - F_{k-1} \times F_k = (-1)^k.$$

Prova: A prova desta propriedade será feita por indução matemática.

Ao substituir, na igualdade $F_{k-2} \times F_{k+1} - F_{k-1} \times F_k = (-1)^k$, $k = 2$, obtemos, sucessivamente, as igualdades

$$F_0 \times F_3 - F_1 \times F_2 = (-1)^2,$$

$$1 \times 3 - 1 \times 2 = (-1)^2,$$

$$1 = 1.$$

O que mostra que a igualdade $F_{k-2} \times F_{k+1} - F_{k-1} \times F_k = (-1)^k$ se verifica para $k = 2$.

Seguidamente, vamos admitir que para um certo número natural $m \geq 2$ se tem $F_{m-2} \times F_{m+1} - F_{m-1} \times F_m = (-1)^m$ — hipótese de indução.

Temos, então, que mostrar que $F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = (-1)^{m+1}$. Tendo presente a forma recursiva de obtenção dos números de Fibonacci, bem como a hipótese de indução,

obtem-se sucessivamente

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = F_{m-1} \times (F_{m+1} + F_m) - (F_{m-1} + F_{m-2}) \times F_{m+1},$$

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = F_{m-1} \times F_{m+1} + F_{m-1} \times F_m - F_{m-1} \times F_{m+1} - F_{m-2} \times F_{m+1},$$

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = F_{m-1} \times F_m - F_{m-2} \times F_{m+1},$$

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = -(F_{m-2} \times F_{m+1} - F_{m-1} \times F_m),$$

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = -(-1)^m,$$

$$F_{m-1} \times F_{m+2} - F_m \times F_{m+1} = (-1)^{m+1}.$$

Depois de termos verificado a validade da propriedade, em discussão, para $k = 2$, a demonstração da sua hereditariedade, permite-nos concluir a prova, por aplicação do princípio de indução matemática. \square

Lema A.5 *Sejam $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci e R_1, R_2, \dots, R_n , n números reais ($n \geq 2$) tais que $R_{k+1} = \frac{1}{R_k} - 1$, $k = 1, 2, \dots, n-1$, e $\frac{1}{2} \leq R_k \leq 1$, $k = 1, 2, \dots, n$.*

As seguintes condições são equivalentes:

- $R_1 \times R_2 \times \dots \times R_n = \frac{1}{F_{n+1}};$
- $R_1 = \frac{F_n}{F_{n+1}}.$

Prova: Suponha-se que $R_1 = \frac{F_n}{F_{n+1}}$. Vamos provar que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$. Como por hipótese $R_1 = \frac{F_n}{F_{n+1}}$, aplicando a Igualdade (A.6) (pág. 151), resulta

$$\prod_{i=1}^n R_i = (-1)^n \times \left(F_{n-2} - F_{n-1} \times \frac{F_n}{F_{n+1}} \right),$$

$$\prod_{i=1}^n R_i = (-1)^n \times \frac{F_{n-2} \times F_{n+1} - F_{n-1} \times F_n}{F_{n+1}},$$

$$\prod_{i=1}^n R_i = \frac{1}{F_{n+1}} \quad (\text{atendendo ao Lema A.4}).$$

Admitamos, agora, que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$. Para concluir a prova do Lema, temos que provar que, sob esta hipótese, $R_1 = \frac{F_n}{F_{n+1}}$. Atendendo à Igualdade (A.6) (pág. 151), obtém-se

$$\prod_{i=1}^n R_i = (-1)^n \times (F_{n-2} - F_{n-1} \times R_1).$$

Então,

$$(-1)^n \times (F_{n-2} - F_{n-1} \times R_1) = \frac{1}{F_{n+1}},$$

$$(-1)^n \times F_{n-2} \times F_{n+1} - (-1)^n \times F_{n-1} \times F_{n+1} \times R_1 = 1,$$

$$R_1 = \frac{1 - (-1)^n \times F_{n-2} \times F_{n+1}}{-(-1)^n \times F_{n-1} \times F_{n+1}},$$

$$R_1 = \frac{1 - (-1)^n \times [(-1)^n + F_{n-1} \times F_n]}{-(-1)^n \times F_{n-1} \times F_{n+1}} \quad (\text{atendendo ao Lema A.4}),$$

$$R_1 = \frac{1 - 1 - (-1)^n \times F_{n-1} \times F_n}{-(-1)^n \times F_{n-1} \times F_{n+1}},$$

$$R_1 = \frac{F_n}{F_{n+1}}.$$

□

A partir de agora, estamos em condições de concretizar o nosso propósito, o qual passa por mostrar que a solução ótima do Problema (A.3) é $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$. De facto, faremos mais do que isso: mostraremos que esta solução ótima é única.

Teorema A.1 *Sejam:*

- $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci;
- R_1, R_2, \dots, R_n , n ($n \geq 2$) números reais tais que $R_{k+1} = \frac{1}{R_k} - 1$, $k = 1, 2, \dots, n-1$,
e $\frac{1}{2} \leq R_k \leq 1$, $k = 1, 2, \dots, n$.

Então, $R_1 \times R_2 \times \cdots \times R_n \geq \frac{1}{F_{n+1}}$. Além disso, $R_1 \times R_2 \times \cdots \times R_n = \frac{1}{F_{n+1}}$ se, e só se, $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$.

Prova: Atendendo à hipótese e ao Lema A.1, para $k \geq 2$, $R_k = -\frac{F_{k-2} - F_{k-1} \times R_1}{F_{k-3} - F_{k-2} \times R_1}$. Então,

$$\begin{aligned} R_1 \times R_2 \times \cdots \times R_n &= R_1 \times R_2 \times \left[-\frac{F_1 - F_2 \times R_1}{F_0 - F_1 \times R_1} \right] \times \left[-\frac{F_2 - F_3 \times R_1}{F_1 - F_2 \times R_1} \right] \times \cdots \\ &\quad \cdots \times \left[-\frac{F_{n-3} - F_{n-2} \times R_1}{F_{n-4} - F_{n-3} \times R_1} \right] \times \left[-\frac{F_{n-2} - F_{n-1} \times R_1}{F_{n-3} - F_{n-2} \times R_1} \right], \end{aligned}$$

$$R_1 \times R_2 \times \cdots \times R_n = R_1 \times \left(\frac{1}{R_1} - 1 \right) \times (-1)^{n-2} \times \frac{F_{n-2} - F_{n-1} \times R_1}{1 - 1 \times R_1},$$

$$R_1 \times R_2 \times \cdots \times R_n = (-1)^n \times (F_{n-2} - F_{n-1} \times R_1), \quad (\text{A.6})$$

$$R_1 \times R_2 \times \cdots \times R_n = (-1)^n \times F_{n-2} + F_{n-1} \times R_1 \times (-1)^{n+1}.$$

Do Lema A.3, sabe-se que para $n \geq 2$, $(-1)^{n+1} \times R_1 \geq (-1)^{n+1} \times \frac{F_n}{F_{n+1}}$. Logo,

$$R_1 \times R_2 \times \cdots \times R_n \geq (-1)^n \times F_{n-2} + F_{n-1} \times (-1)^{n+1} \times \frac{F_n}{F_{n+1}},$$

$$R_1 \times R_2 \times \cdots \times R_n \geq (-1)^n \times \left(F_{n-2} - F_{n-1} \times \frac{F_n}{F_{n+1}} \right),$$

$$R_1 \times R_2 \times \cdots \times R_n \geq (-1)^n \times (F_{n-2} \times F_{n+1} - F_{n-1} \times F_n) \times \frac{1}{F_{n+1}}.$$

Atendendo ao Lema A.4, para $n \geq 2$, $F_{n-2} \times F_{n+1} - F_{n-1} \times F_n = (-1)^n$. Por conseguinte,

$$R_1 \times R_2 \times \cdots \times R_n \geq (-1)^n \times (-1)^n \times \frac{1}{F_{n+1}},$$

$$R_1 \times R_2 \times \cdots \times R_n \geq \frac{1}{F_{n+1}}.$$

Falta mostrar que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$ é equivalente a $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$. Suponha-se que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$. Atendendo ao Lema A.5, $R_1 = \frac{F_n}{F_{n+1}}$.

Usando o método de indução, vamos mostrar que, nestas condições, $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$. Se $k = 1$, obtém-se $R_1 = \frac{F_n}{F_{n+1}}$, o que é verdade atendendo à hipótese. Admitamos, que para um certo número natural m se tem $R_m = \frac{F_{n-m+1}}{F_{n-m+2}}$. Temos que mostrar que $R_{m+1} = \frac{F_{n-m}}{F_{n-m+1}}$. Como, por hipótese, $R_{k+1} = \frac{1}{R_k} - 1$, então tem-se

$$R_{m+1} = \frac{1}{R_m} - 1,$$

$$R_{m+1} = \frac{1}{\frac{F_{n-m+1}}{F_{n-m+2}}} - 1,$$

$$R_{m+1} = \frac{F_{n-m+2} - F_{n-m+1}}{F_{n-m+1}},$$

$$R_{m+1} = \frac{F_{n-m}}{F_{n-m+1}} \quad (\text{atendendo à definição de número de Fibonacci}).$$

Para finalizar a prova, suponhamos que $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, com $k = 1, 2, \dots, n$. Temos que provar que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$. Fazendo $k = 1$, na igualdade $R_k = \frac{F_{n-k+1}}{F_{n-k+2}}$, obtém-se $R_1 = \frac{F_n}{F_{n+1}}$. Atendendo ao Lema A.5, conclui-se que $\prod_{i=1}^n R_i = \frac{1}{F_{n+1}}$. \square

A forma recursiva (A.1) como se definiu a sucessão de Fibonacci gera a dificuldade de não se possível calcular um número de Fibonacci sem conhecer os dois anteriores. A proposição seguinte garante um termo geral para a sucessão de Fibonacci, o que resolve a dificuldade atrás referida.

Proposição A.1 (Fórmula de Binet) *Seja $(F_n)_{n \in \mathbb{N}_0}$ a sucessão de Fibonacci. Então,*

para todo $n \in \mathbb{N}_0$,

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right]. \quad (\text{A.7})$$

Prova: Vamos recorrer ao método de indução para realizar a prova.

Para $n = 0$, da Fórmula (A.7), resulta

$$\begin{aligned} F_0 &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{0+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{0+1} \right], \\ F_0 &= \frac{1}{\sqrt{5}} \times \frac{2\sqrt{5}}{2}, \\ F_0 &= 1, \end{aligned}$$

o que é verdade, atendendo à Definição (A.1).

Para $n = 1$, da Fórmula (A.7), resulta

$$\begin{aligned} F_1 &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^2 - \left(\frac{1-\sqrt{5}}{2} \right)^2 \right], \\ F_1 &= \frac{1}{\sqrt{5}} \times \frac{4\sqrt{5}}{4}, \\ F_1 &= 1, \end{aligned}$$

o que é verdade, atendendo à Definição (A.1).

Suponha-se, agora, que para dois números (inteiros não negativos) consecutivos $k-2$ e $k-1$ se tem

$$F_{k-2} = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k-1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \right] \quad (\text{A.8})$$

e

$$F_{k-1} = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right]. \quad (\text{A.9})$$

Estabelecida a hipótese de indução (igualdades (A.8) e (A.9)), torna-se necessário mostrar que

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k+1} \right].$$

Da Fórmula (A.1), que define a sucessão de Fibonacci, e atendendo à hipótese de indução (A.8) e (A.9), conclui-se que

$$F_k = F_{k-2} + F_{k-1},$$

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k-1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \right] + \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right],$$

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k-1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} + \left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right],$$

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k \left(1 + \left(\frac{1+\sqrt{5}}{2} \right)^{-1} \right) - \left(\frac{1-\sqrt{5}}{2} \right)^k \left(1 + \left(\frac{1-\sqrt{5}}{2} \right)^{-1} \right) \right],$$

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k \left(\frac{1+\sqrt{5}}{2} \right) - \left(\frac{1-\sqrt{5}}{2} \right)^k \left(\frac{1-\sqrt{5}}{2} \right) \right],$$

$$F_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{k+1} \right].$$

Fica assim garantida a hereditariedade da propriedade, o que permite terminar a prova. \square

Apêndice B

Determinante da Matriz de Vandermonde

Neste apêndice será discutida a não singularidade da matriz de Vandermonde referida na Subsecção 3.1.1. Para o efeito, foi escolhido o autor [24, Monteiro] para referência.

Na matriz quadrada $(n + 1) \times (n + 1)$

$$V_n = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \quad (\text{B.1})$$

os elementos das suas linhas estão em progressão geométrica. Por esta razão, diz-se que é uma matriz de Vandermonde¹. Para provarmos que o seu determinante é não nulo se e só se $x_i \neq x_j$ para $i \neq j$, faremos uso do Teorema de Laplace, razão pela qual apresentamos a seguir o seu enunciado.

Teorema B.1 (Teorema de Laplace) *Seja A uma matriz quadrada de ordem n com $n \geq 2$, o determinante de A é dado pela soma dos produtos dos elementos de uma fila (linha ou coluna) qualquer pelos respectivos co-fatores, isto é:*

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (\text{B.2})$$

¹A matriz de Vandermonde pode não ser quadrada. No entanto, apenas nos interessam as que são quadradas.

então

$$\det A = \sum_{j=1}^n a_{ij} A_{ij}, \quad \text{sendo } i \text{ uma linha qualquer da matriz}$$

ou

$$\det A = \sum_{i=1}^n a_{ij} A_{ij}, \quad \text{sendo } j \text{ uma coluna qualquer da matriz.}$$

A proposição seguinte garante que uma matriz quadrada de Vandermonde é não singular, caso seja gerada pelo vetor (x_0, x_1, \dots, x_n) de componentes distintas.

Proposição B.1 *Se $x_i \neq x_j$ para $i \neq j$, então,*

$$\det \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} = \prod_{0 \leq i < j \leq n} (x_j - x_i) \neq 0. \quad (\text{B.3})$$

Prova: Apresentaremos uma prova por indução finita sobre n . Para $n = 1$, tem-se

$$\det V(x_0, x_1) = \begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} = x_1 - x_0.$$

Portanto, a propriedade é válida para $n = 1$.

Suponhamos que a propriedade é válida para $n - 1$ e mostremos que é válida para n . Seja

$$V(x_0, x_1, \dots, x_n) = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix}$$

a matriz de Vandermonde de ordem n . Multiplicando a primeira coluna por x_0 e subtraindo o resultado à segunda coluna, multiplicando a segunda coluna por x_0 e subtraindo o resultado a terceira coluna e assim sucessivamente, obtemos

$$\det V(x_0, x_1, \dots, x_n) = \det \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & x_1(x_1 - x_0) & \cdots & x_1^{n-1}(x_1 - x_0) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & x_n(x_n - x_0) & \cdots & x_n^{n-1}(x_n - x_0) \end{bmatrix}.$$

Pelo Teorema de Laplace e sabendo que ao multiplicar uma fila qualquer de uma matriz quadrada M de ordem n por um real k , o determinante da nova matriz M' obtida será o produto de k pelo determinante de M , temos então que

$$\det V(x_0, x_1, \dots, x_n) = (x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0) \det \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}.$$

Assim, podemos escrever

$$\det V(x_0, x_1, \dots, x_n) = (x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0) \det V(x_1, x_2, \dots, x_n)$$

$$= \prod_{j=1}^n (x_j - x_0) \det V(x_1, x_2, \dots, x_n).$$

Mas $\det V(x_1, x_2, \dots, x_n)$ é por hipótese de indução o determinante de Vandermonde de ordem $n - 1$. Assim, efetuando o produto, obtemos o determinante de ordem n de Vandermonde,

$$\det V(x_0, x_1, \dots, x_n) = \prod_{0 \leq i < j \leq n} (x_j - x_i).$$

Ora, atendendo a que $x_i \neq x_j$ para $i \neq j$, fica provado que $\det V(x_0, x_1, \dots, x_n) \neq 0$. \square

Bibliografia

- [1] A. Antoniou and W. Lou. *Practical Optimization, Algorithms and Engineering Applications*. Springer, New York, 2007.
- [2] R. S. Biryukov. *Methods of Optimization*. Lobachevsky State University of Nizhni Novgorod, Russian Federation, 2011. Notas de aula.
- [3] Domingos Cardoso. *Tópicos de Optimização Não Linear*. Universidade de Aveiro, 1999. Notas de aula.
- [4] Steven C. Chapra and Raymond P. Canale. *Numerical Methods for Engineers*. McGraw-Hill, New York, 6th edition, 2010.
- [5] Edwin K. P. Chong and Stanislaw H. Żak. *An Introduction to Optimization, 2nd. Edition*. John Wiley & Sons, New York, 2001.
- [6] Edwin K. P. Chong and Stanislaw H. Żak. *An Introduction to Optimization, 4th. Edition*. John Wiley & Sons, New York, 2013.
- [7] Gabinete de Avaliação Educacional. *Matemática — Questões de Exame do 12.º ano 1997 – 2005*. Editorial do Ministério da Educação, Lisboa, 2005.
- [8] Doreen De Leon. *Single-Variable Unconstrained Optimization*. Department of Mathematics, California State University, 2012. Notas de aula.
- [9] Francisco Gilberto de Sousa Carvalho. *Algoritmo do Ponto Proximal Generalizado em Espaços de Hilbert para o Problema de Desigualdade Variacional*. Universidade Federal do Piauí, 2010. Dissertação de mestrado.
- [10] José Joaquim Magalhães de Sousa Pinto. *Curso de Análise Matemática*. Universidade de Aveiro, Aveiro, 2010.
- [11] Jorge Sá Esteves. *Lições de Optimização Numérica*. Universidade de Aveiro, 2012. Notas de aula.
- [12] Edite Manuela da G. P. Fernandes. *Computação Numérica*. Universidade do Minho, Braga, 1997.

- [13] Mário S. R. Figueira. *Fundamentos de Análise Infinitesimal*. Faculdade de Ciências da Universidade de Lisboa, Lisboa, 2001.
- [14] Roger Fletcher. *Practical Methods of Optimization, Vol. 1*. John Wiley & Sons, New York, 1980.
- [15] Roger Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, 2nd edition, 2008.
- [16] Hansen, Pierre, Jaumard, Brigitte and Lu, Shi-Hui. Global Optimization of Univariate Lipschitz Functions: Survey and Properties. *Mathematical Programming*, 55, 1992.
- [17] Floyd Hanson. *Golden Section Search Method*. Department of Mathematics, Statistics, and Computer Science. University of Illinois at Chicago, 2004. Notas de aula.
- [18] Desmond J. Higham and Nicholas J. Higham. *MATLAB Guide*. Society for Industrial and Applied Mathematics, University City Science Center, Philadelphia, 2nd edition, 2005.
- [19] Jaan Kiusalaas. *Numerical Methods in Engineering With MATLAB*. Cambridge University Press, Cambridge, UK, 2005.
- [20] Andrew Knight. *Basics of MATLAB and Beyond*. Chapman & Hall, 2000.
- [21] José Donizetti Lima. *Elementos Básicos de Programação em MATLAB*. Ministério da Educação, Centro Federal de Educação Tecnológica do Paraná, Unidade de Pato Branco. Notas.
- [22] Joe LoVetri. *Optimization Methods*. Faculty of Engineering, University of Monitoba, Canada, 2001. Notas de aula.
- [23] John H. Mathews and Kurtis K. Fink. *Numerical Methods Using Matlab*. Prentice-Hall, Upper Saddle River, New Jersey, 4th edition, 2004.
- [24] Agostinho Jorge Tavares Monteiro. *Otimização Não Linear de Mínimos Quadrados*. Universidade de Aveiro, 2013. Dissertação de mestrado.
- [25] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 1999.
- [26] Singiresu S. Rao. *Engineering Optimization, Theory and Practice*. John Wiley & Sons, Hoboken, New Jersey, 4th edition, 2009.
- [27] Vírginia Santos. *Cálculo I, Cálculo com uma Variável Real*. Departamento de Matemática da Universidade de Aveiro, Aveiro, 2009. Notas de aula.
- [28] Stongin Roman G. Sergeyev, Yaroslav D. and Daniela Lera. *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York, 2013.

- [29] Sergeyev, Yaroslav D. and Lera, Daniela. Accelerating of Univariate Global Optimization Algorithms Working With Lipschitz Functions and Lipschitz First Derivative. *SIAM Journal of Optimization*, 23, 2013.
- [30] James Stewart. *Calculus Early Transcendentals*. Thomson Books/Cole, Belmont, , California, 6th edition, 2008.
- [31] A. L. Peressini; F. E. Sullivan and J. J. Uhl Jr. *The Mathematics of Nonlinear Programming*. Springer, New York, 1988.
- [32] Earl W. Swokowski. *Cálculo com Geometria Analítica, Vol. 1*. McGraw-Hill, São Paulo, 1983.
- [33] P. Venkataraman. *Applied Optimization with MATLAB Programming, 2nd. Edition*. John Wiley & Sons, Hoboken, N.J., 2009.
- [34] Douglass J. Wilde. *Optimum Seeking Methods*. Prentice-Hall, Englewood Cliffs, N.J., 1964.
- [35] Xin-She Yang. *Introduction to Mathematical Optimization*. Cambridge International, Cambridge, UK, 2008.
- [36] Anatoly Zhihiglavsky and Antanas Žilinskas. *Stochastic Global Optimization*. Springer, New York, 2008.